# NAVAL POSTGRADUATE SCHOOL

## Monterey, California



# THESIS

IS HYPERTEXT A SOLUTION TO
IMPLEMENTING AN ADP SECURITY PROGRAM IN DON?
ISSUES AND PROBLEMS

by

Robert Andrew VanMeter

September, 1991

Thesis Advisor: Tung Xuan Bui

Approved for public release; distribution is unlimited

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

# REPORT DOCUMENTATION PAGE

| | |
|---|---|
| 1a REPORT SECURITY CLASSIFICATION<br>UNCLASSIFIED | 1b RESTRICTIVE MARKINGS |
| 2a SECURITY CLASSIFICATION AUTHORITY | 3 DISTRIBUTION/AVAILABILITY OF REPORT<br>Approved for public release; distribution is unlimited. |
| 2b DECLASSIFICATION/DOWNGRADING SCHEDULE | |
| 4 PERFORMING ORGANIZATION REPORT NUMBER(S) | 5 MONITORING ORGANIZATION REPORT NUMBER(S) |
| 6a NAME OF PERFORMING ORGANIZATION<br>Naval Postgraduate School<br>6b OFFICE SYMBOL *(If applicable)* AS | 7a NAME OF MONITORING ORGANIZATION<br>Naval Postgraduate School |
| 6c ADDRESS *(City, State, and ZIP Code)*<br>Monterey, CA 93943-5000 | 7b ADDRESS *(City, State, and ZIP Code)*<br>Monterey, CA 93943-5000 |
| 8a NAME OF FUNDING/SPONSORING ORGANIZATION<br>8b OFFICE SYMBOL *(If applicable)* | 9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER |
| 8c ADDRESS *(City, State, and ZIP Code)* | 10 SOURCE OF FUNDING NUMBERS<br>Program Element No / Project No / Task No / Work Unit Accession Number |

11 TITLE *(Include Security Classification)*
IS HYPERTEXT A SOLUTION TO IMPLEMENTING AN ADP SECURITY PROGRAM IN DON? ISSUES AND PROBLEMS

12 PERSONAL AUTHOR(S) VanMeter Robert A.

| 13a TYPE OF REPORT | 13b TIME COVERED | 14 DATE OF REPORT *(year, month, day)* | 15 PAGE COUNT |
|---|---|---|---|
| Master's Thesis | From To | 1991, SEPTEMBER | 141 |

16 SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

| 17 COSATI CODES | | | 18 SUBJECT TERMS *(continue on reverse if necessary and identify by block number)* |
|---|---|---|---|
| FIELD | GROUP | SUBGROUP | Hypertext, Hypermedia, CAI, ADP Security |

19 ABSTRACT *(continue on reverse if necessary and identify by block number)*

The goal of this thesis is to provide an overview of hypertext to determine its feasibility for resolving some of the problems currently facing newly assigned and inexperienced ADP security officers. The proclivity within DOD for using documents in virtually every facet of work suggests that hypertext has a promising future in the DOD. To implement an ADP security program in the Navy, the information presented in the DON AIS Security Guidelines should be carefully selected and filtered to derive a tool that provides an effective and circumstance-shaped source of information, guidance and reference. From a design standpoint, it is important to integrate hypertext technology with other computer based tools -- such as expert systems and simulation models -- to fully exploit the potential of this new technology.

| | | |
|---|---|---|
| 20 DISTRIBUTION/AVAILABILITY OF ABSTRACT<br>☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS REPORT ☐ DTIC USERS | 21 ABSTRACT SECURITY CLASSIFICATION<br>Unclassified | |
| 22a NAME OF RESPONSIBLE INDIVIDUAL<br>Tung Xuan Bui | 22b TELEPHONE *(Include Area code)*<br>408 646 2630 | 22c OFFICE SYMBOL<br>AS/Bd |

**DD FORM 1473, 84 MAR** 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

Approved for public release; distribution is unlimited

# Is Hypertext a Solution to Implementing an ADP Security Program in DON? Issues and Problems

by

Robert Andrew VanMeter
Lieutenant Commander, United States Naval Reserve
B.A., University of Rochester

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
September, 1991

Author: Robert A. VanMeter

Approved by: Tung Xuan Bui, Thesis Advisor

Moshe Zviran, Second Reader

David R. Whipple, Chairman
Department of Administrative Sciences

## ABSTRACT


The goal of this thesis is to provide an overview of hypertext to determine its feasibility for resolving some of the problems currently facing newly assigned and inexperienced ADP security officers. The proclivity within DOD for using documents in virtually every facet of work suggests that hypertext has a promising future in the DOD. To implement an ADP security program in the Navy, the information presented in the DON AIS Security Guidelines should be carefully selected and filtered to derive a tool that provides an effective and circumstance-shaped source of information, guidance and reference. From a design standpoint, it is important to integrate hypertext technology with other computer based tools -- such as expert systems and simulation models -- to fully exploit the potential of this new technology.

## TABLE OF CONTENTS

# I. INTRODUCTION

## A. OBJECTIVES

The objective of this thesis is to determine the feasibility of hypertext for resolving some of the problems currently facing newly assigned ADP security officers throughout the Department of the Navy.

A secondary objective is to identify issues and problems with implementing hypertext technology within the Department of Defense. The intent is to provide a solid background for evaluating potential hypertext applications and to identify some areas where the technology might be especially useful.

## B. BACKGROUND

Naval Officers are often assigned collateral duties outside their area of professional expertise. An example of this would be an aviator assigned the collateral duty of ADP Security Officer. The collateral duty requires extensive knowledge and much work to effectively implement. The officer has little time to spend working in that capacity and their only guidance may be a very intimidating instruction.

There are a multitude of problems which face the newly assigned ADP Security Officer. Some of these are attributable to a general lack of computer expertise coupled with the problems inherent with the primary reference instruction. The

primary reference for implementing an ADP security program in the Navy is the Department of the Navy Automated Information Systems Security Guidelines (DON AIS Security Guidelines). This is a very extensive and comprehensive manual designed to address all the issues associated with ADP security program implementation.

Although the idea of hypertext has been around for over 45 years, it has just recently emerged as an area of great research interest. "The reason people are getting excited about hypertext now even though the concept dates back to 1945 is that it can now be implemented with commercially used technology." [Nielsen, 1990a] Vast improvements in technology have begun to make hypertext a realistic solution to many different problems. Hypertext is exciting because it is said to mimic the way humans think. It allows information to be organized and accessed in many different ways according to the reader's needs and desires.

Hypertext applications are based on documents and information. Because the Navy's ADP security program is implemented via a document, transforming this manual into a hypertext environment might help resolve some of the problems which face a newly assigned and inexperienced ADP Security Officer. Hypertext might have many other useful applications within the Department of Defense as well.

Before blindly applying new technology such as hypertext to a particular problem, a thorough understanding of the

technology's capabilities and limitations is required. How this technology relates to the problem domain must be carefully established. Hypertext offers many very exciting possibilities for applications within the Department of Defense.

## C. THE RESEARCH QUESTION

The research question is "can hypertext technology resolve the problems associated with implementing an ADP security program in the Department of the Navy?"

## D. SCOPE AND LIMITATIONS

The thesis has three basic parts. The first part identifies some of the problems facing newly assigned ADP Security Officers. The second part of the thesis gives an extensive overview of hypertext to include its capabilities, limitations, and design issues. The last part of the thesis focuses on the general application of hypertext within the Department of Defense and the specific problem of designing a system which addresses the research question. The thesis does not provide design specifications for this system but rather, addresses some of the design considerations and functional characteristics of a potential system.

This thesis additionally identifies potential useful application areas for hypertext within the Department of

Defense. It does not recommend specific systems but instead focuses on general areas of beneficial employment.

## E. RESEARCH METHODOLOGY

A literature review was conducted to identify the capabilities, limitations, and design considerations for hypertext system. Additional literature review was conducted in the areas of expert system technology and computer aided instruction to assess the benefit of their incorporation into hypertext systems.

This background established the framework to identify potential hypertext applications in the Department of Defense and specifically analyze hypertext feasibility for resolving ADP security program implementation problems

## F. ORGANIZATION OF THE STUDY

A background scenario is presented in Chapter II to help identify problems involved with implementing an ADP security program in a Navy command. Specific issues and possible solutions are identified. A hypertext system is proposed as one possible solution to some of the problems identified in the scenario. Chapter III provides a general overview of hypertext including its basic architecture, capabilities, potential applications, potential weaknesses. Chapter IV looks at the design issues involved with building a hypertext system. Issues of useability, database construction,

navigation and information retrieval, and authoring are reviewed. This chapter also provides a comparison of hypertext to existing computer systems and printed material. Chapter V reviews potential applications in the Department of Defense. Chapter VI evaluates the practicality of a hypertext solution to address some of the issues identified in Chapter II. This chapter al o identifies characteristics and features of a potential hypertext system. Chapter VII provides a summary and conclusions of the research and recommends possible topics for future study.

# II. BACKGROUND

## A. INTRODUCTION

The following scenario is based on a real life experience faced by the author during a previous tour of duty. The author believes this scenario to fairly represent most of the issues facing many new ADP Security Officers throughout the Navy. This background is the catalyst for this thesis. The immense frustration experienced by the author when trying to develop an ADP security program for his command led to a search for a means to assist those small commands which possess limited computer expertise. This search led to a review of expert system technology, computer assisted instruction, and hypertext technology and their possible integration for possible assistance with this problem.

## B. A NEW SECURITY OFFICER'S DILEMMA

Lieutenant Smith reported to his current duty station in February 1987. This was his first tour as a Naval Reserve Officer on active duty in a naval reserve patrol aviation squadron. LT Smith had served previous tours in an active duty patrol aviation squadron and in the training command. After an initial indoctrination into the squadron and some training concerning the training and administration of naval

reserve personnel, he was assigned as the squadron Administrative Department Officer. The squadron is manned by six officers and approximately 100 enlisted personnel who are responsible for the roughly 200 drilling reservists which complete the command manning structure. The squadron is manned with enlisted rates required to perform aviation maintenance, and operations. The officers are all aviators; either pilots or Naval Flight Officers. Despite being a Naval Reserve command, the squadron is required to maintain all of the programs and administrative requirements of an active duty squadron manned to the same level with all active duty personnel. The TAR officer will assume many collateral duties which may be a full time job for his counterpart in an active duty command. As implied, Lt Smith had many collateral duties aside from his primary duties as administrative officer.

One morning, during the course of reading the routine message traffic for the day, Lt Smith came across an action message from the Commander Naval Reserve Force concerning ADP security. Essentially the message said that the majority of commands were not in compliance with the requirements outlined in OPNAVINST 5239.1A which was dated Aug of 1983. The message directed all deficient commands to implement an ADP security program ASAP and outlined specific reporting requirements for those commands to ensure compliance. This message immediately caught the attention of LT Smith who recognized that the command was in fact deficient and did not even have an ADP

security officer designated. He also recognized, given the nature of his collateral duties, he was a likely candidate for this job. After some discussion with the Commanding Officer and other department heads, LT Smith was designated as the command ADP security officer and subsequently tasked with responding to this message. The task seemed to have additional urgency because the command had just recently received six new Z-248 computers which constituted the bulk of its computer assets.

LT Smith was now faced with a dilemma. How was he going to implement a program for which he had no training or previous experience? In fact LT Smith's experience with computers was so limited that he had accidently reformatted the hard drive on the Administrative department's only computer just two months prior. Actually this is a common circumstance in many jobs within the Navy. However, this case was slightly different from LT Smith's previous experience in that this program seemed to require experience and knowledge that could not easily be picked up through on the job training. Additionally, LT Smith's other duties did not allow an inordinate amount of time to deal with the problem.

After review of the OPNAVINST 5239.1A, LT Smith was overwhelmed by both the volume and complexity of the program. The instruction itself was full of acronyms and terminology completely unfamiliar to him. Many of the acronyms were so similar (e.g., ADPSO, ADPSSO, ADP, AAS, AADPSP, OISSO, TASO,

OIS) they caused great confusion when trying to interpret the instruction. Even the spelled out terminology seemed less that straight forward. The concepts of an Activity Accreditation Schedule, Risk Assessment, and System Test and Evaluation were difficult to grasp and seemed like too much effort for the limited computer assets owned by the command. It seemed that the instruction had been written for a very large command which possessed a large number of computer assets including mainframes and minicomputers. These commands had personnel with computer backgrounds trained to deal with the issues raised by the instruction. The instruction seemed to completely ignore the myriad of small commands scattered throughout the navy which possessed limited computer assets and very limited personnel assets capable of dealing with these issues.

The next logical step for LT Smith was to get the training needed to complete his assigned task. He additionally sought to determine resident command computer expertise to assign collateral duties designated by the instruction to implement the program. The most computer literate person in the command was the Command Master Chief. LT Smith quickly arranged for himself and the master chief to attend the next open ADP security officers course taught by the nearest available Navy Regional Data Automation Center (NARDAC). LT Smith was sure that given the training and the assistance of the Command

Master Chief, he would be able to effectively implement an ADP security program for his command.

The training was as overwhelming as the original review of the OPNAV instruction. The school familiarized LT Smith with terminology, acronyms, the general principles and procedures outlined by the OPNAV instruction, and gave him a better appreciation for the very real need for effective ADP security measures. However, the course seemed to be oriented towards the major commands. It only briefly touched on the issues germane for small commands. For example there are two Risk Assessment methodologies, one of which is more applicable for the small command. However, the course spent the majority of time teaching the more complex methodology designed for the commands with substantial computer resources. Additionally, the course highlighted an even greater volume of source information that needed to be sifted through to glean the important points and effectively implement the program. The course further emphasized that the program was very cumbersome indeed, with no easy way out for those commands with only limited computer resources and limited personnel computer expertise. LT Smith and the Master Chief returned to the command with a huge course notebook full of additional reference material and a full appreciation for the genuine difficulties that faced them when trying to respond to the original action message.

At this point LT Smith felt he was almost no better off than when he started. It was three months later, and the deadline for message response was quickly approaching. The amount of time required to implement the program in accordance with the OPNAVINST 5239.1A was very substantial and there were many other aspects of his job that seemed more immediate and worthwhile. LT Smith recognized the importance of implementing effective ADP security measures. On the other hand he felt that the huge volume of paperwork requisite for activity accreditation was not worth the effort for the limited computer resources available to the command. LT Smith realized that he had to establish some priorities. His goal now was to implement the security measures which were most practical and most beneficial to the routine operations of the command and try to catch up on the official paperwork when time allowed.

Here again, LT Smith faced another roadblock of sorts. He felt that he did not have the technical expertise available to determine the most practical and beneficial computer security measures to be implemented. Certain procedures seemed relatively obvious and straight forward to implement. Examples included establishing backup procedures, instituting rules concerning food and drinks near computers, and installing surge protectors and static mats. The benefit of other measures such as password protection schemes and virus detection and prevention procedures seemed less obvious.

Lastly LT Smith recognized that the fundamental problem for both himself and the command was a marked deficiency in computer expertise. This pointed out the need for training. "A large portion of damage to computers is unintentional and non-malicious from untrained personnel." [Evans, 1990] (Remember LT Smith reformatted the hard drive!) If only the budget allowed for all the training that LT Smith felt was necessary!

The original message had directed an initial response requesting a schedule of milestones for program implementation. The initial response was sent with the requested schedule. LT Smith projected the milestone completion dates based on the no later than completion dates directed in the action message. This was also done immediately prior to attending the ADP security officer's course. At that time LT Smith still assumed that he would be able to meet the deadlines directed once he received some formal training. No further messages were ever sent. None of the published milestones were met and no follow up messages directing further action were ever received. Several months later the action message was all but forgotten, lost in a myriad of other projects and details. Despite implementation of several security measures, ADP security at the command was in a general state of neglect. The command later passed a major Command Inspection with no major discrepancies. No further action was ever taken during the remainder of LT Smith's tour.

## C. ISSUES AND RECOMMENDATIONS

There are multiple issues posed by the background scenario which face many newly assigned ADP security officers. The following bullets briefly outline these issues.

- Many Navy commands face a shortage of personnel with any type of computer background. Command manning documents do not provide for computer specialists for the majority of operational commands. Command computer expertise is often limited to a few 'hackers' with only home micro computer experience. This experience does not provide the requisite background to deal with the problems posed by the implementation of an effective ADP security program.

- Computer Security is often not given highest priority, either by top management or by the personnel designated to implement the program. There are often conflicting priorities which receive attention first.

- Computers are still a relative novelty for many navy commands, especially the small ones. Distributed computing has been in many commands for less than five years. Problems which arise through working with computers are only recently being recognized as significant.

- Instructions used as references are complicated and full of acronyms and jargon unfamiliar to computer novices. Procedures detailed in these instructions are complex and require extensive training to correctly perform.(e.g., Risk Assessment) These instructions require computer literacy and expertise not found in great supply in small Navy commands.

- Training which is available is not tailored to individual command needs and can be more complex than required. Even if training is available, commands may be unwilling to take full advantage of it due to austere budget constraints.

- Computer security is important to the routine operations of most commands yet effective security measures are often not implemented and incorporated as part of those routine operations.

The nature of the issues presented requires multiple actions to effectively address. Many of these issues have been

recognized by major claimants and some corrective actions have been taken. Many major claimants have sample ADP security programs that can be used as a model for the activities under their claimancy. These programs can be used in a kind of cook book approach to help commands with limited computer expertise implement an ADP security program. Figure 2.1 outlines a sample of one these programs provided by CINCLANTFLT. These programs allow for implementation of a command ADP security program in conformance with standards used to inspect those commands. The principle benefit of this approach is that it enables commands to pass inspections given by their major claimant. Additionally, this approach serves to implement some measure of ADP security procedures and places increased command emphasis on their importance.

Although this solution does address many of the issues raised above, it is not a cure all solution. For example, this solution does not effectively address the issue of training. It actually does very little to educate the commands using this approach concerning formal ADP security measures. Because many commands are so deficient in basic computer expertise, this approach may also lead to a program which looks good on paper, but in actuality provides limited ADP security measures which do not comply with the spirit of intent of the OPNAV instruction. For example, the password protection scheme outlined in the command security instruction may be very different from the scheme actually used in the routine

FROM: CINCLANTFLT, CODE N74

TO:________________

SUBJ: SHIPBOARD ADP SECURITY PROGRAM, AN APPROACH

REF: a. OPNAVINST 5239.1A

ENCL: 1. SHIPBOARD ADP/IS SECURITY ORGANIZATION
2. SAMPLE ADP SECURITY INSTRUCTION
3. SAMPLE ADP SECURITY OFFICER APPOINTMENT LETTER
4. SAMPLE ACTIVITY ADP SECURITY PLAN
5. ACTIVITY ACCREDITATION DEFINITION
6. SAMPLE CHECKLIST RISK ASSESSMENT
7. SAMPLE LEVEL 1 SECURITY OPERATING PROCEDURES
8. SAMPLE LEVEL II & III SECURITY OPERATING PROCEDURES
9. SAMPLE ABBREVIATED SECURITY TEST AND EVALUATION
10. SAMPLE INTERIM AUTHORITY TO OPERATE LETTER

ALL NAVY ACTIVITIES ARE REQUIRED BY REF a TO ESTABLISH AN ACTIVITY ADP SECURITY PROGRAM TO ENSURE THAT EACH COMPUTER SYSTEM OPERATING UNDER ITS CONTROL OPERATES WITH AN ACCEPTABLE LEVEL OF RISK. ENCL. 1 THROUGH 10 ARE OFFERED AS TOOLS TO ASSIST AFLOAT...

Figure 2.1 CINCLANTFLT Shipboard Security Program

operations of the command. Lastly this approach sounds simple, but in fact is not command specific, which implies it covers issues not necessarily applicable to, or practical for the command. The program outlined in Figure 2.1 is almost 75 pages long and still only provides an outline of many of the procedures without answering many of the how to questions sure to come up when attempting to implement the program. For example, the Risk Assessment checklist provided gives no guidance on how to properly complete it. This means additional

source documents are required for reference. Questions concerning how to access threat value and how to determine value of the data are not adequately addressed. Another example concerns the ST&E enclosure. This enclosure provides a detailed checklist for evaluating the security measures in place but it leaves the documented test procedures up to the individual commands. How to do these may be less than obvious.

The issue of training is an area where assistance is required. This includes both general computer training and more specific security training. Recommendations for this could encompass a broad spectrum of solutions which might include Computer Assisted Training, monthly computer security newsletters, more courses offered by regional NARDACs, and on-sight training visits. The primary emphasis should be to provide the most training for the fewest dollars. This training must also address the real need for effective ADP security measures which would help give this issue a higher priority than it might currently enjoy.

One of the primary barriers to implementing effective security measures was the reference instruction used for the program. The OPNAVINST 5239.1A is no longer the primary reference for implementing an ADP security program. Research for this thesis uncovered a Department of the Navy ADP Security Guideline instruction which is now the primary reference manual. Additionally since the authors experience in the case, a new SECNAVINST 5239.2 has been issued which

consolidates the security of all DON AISs. The DON ADP Security Guideline is a very comprehensive manual which negates the requirement for much additional reference material. This represents a big improvement over previous instructions. However, even this instruction suffers from being a very large and complex document which is still full of jargon which may be unfamiliar to many novice ADP Security officers. The basic procedures have remained largely unchanged, which implies they are still complicated and difficult for a novice to grasp.

Another possible course of action is development of an automated system that deals with many of the issues raised above. A Decision Support System for implementing the Navy's ADP security program seems a natural outgrowth from many of these problems.

> The decisions involved in establishing an IS security plan are subjective and unstructured. The crucial elements of risk and vulnerability assessment are subject to personal perceptions of threats to information resources, the impact of realized threats, and the probability of their occurrence. ...A decision support tool can, therefore, provide significant guidance to reduce the risks associated with the inadequate security measures. [Zviran et al, 1990]

While the decisions involved in implementing an ADP security program are subjective, much of the information and rules that such a system would be based on are fairly structured. This implies that much of this information could

be codified into a set of if/then rules suitable for an expert system. A rule based DSS or expert system might readily lend itself to instruction and could also be used as a reference to guide personnel unfamiliar with computer security through an ADP security implementation process. This type of system could give very useful guidance concerning complicated procedures such as risk assessment and ST&E. Such a system could also be used to guide commands to implement security measures that comply with the Trusted Computer Security Evaluation Criteria for class C2 required by the Navy by 1992.

The initial intent of this thesis was to develop a prototype expert system designed to meet the needs described above. Here again, research uncovered action which has been undertaken to effect this type of solution. The Information Systems Technology Center in Pearl Harbor Hawaii has very recently (May 1991) developed a decision support system project called Interactive System for AIS Accreditation (ISAAc). This system has been delivered to Naval Computer and Telecommunications Command (NCTC) for evaluation and is unavailable for review at this time. This system seems to address many of the issues raised in the case study but a full evaluation is necessary before any determination can be made.

The wide range of problems cited in the scenario and the wide variety of solutions posed require that some priorities be established. Because several of the issues outlined above are presently being addressed, actions should be taken to

augment those efforts. Areas which seem to stand out include training, and assistance with interpreting the current instruction. Some how-to help for complicated procedures might be required if the IAASc system proves inadequate.

One possible means with which these issues could be addressed is the development of a hypertext system which uses the DON AIS Security Guidelines as the underlying document. Such a system could improve information access and retrieval from the document, provide supplemental information in difficult to understand areas, provide immediate access to a glossary for unfamiliar ADP terms and acronyms, provide access to additional reference material related to the specific task, provide a tutorial lesson in the difficult to understand areas and even offer expert system advise for some procedures. The primary focus of the system should be to make the information available for DON security procedures more accessible and understandable. The system should be more useful than current manual procedures. Prior to building a prototype system, a thorough understanding of general hypertext is required.

# III. OVERVIEW OF HYPERTEXT

## A. DEFINITIONS

### 1. Definition of Hypertext

Hypertext has received a lot of press lately and at this point many readers familiar with some form of hypertext system probably have some opinion of what hypertext is and what it can do. It is very likely that these opinions are numerous and diverse, reflecting the exposure the reader has had to a particular system. There may be some confusion concerning the term hypermedia as well, as this term is often used interchangeably. Hypertext is an extremely broad concept that encompasses a huge spectrum of potential applications.

Perhaps the simplest way to describe hypertext is to contrast it with more familiar forms of text such as books and reference materials. Traditional text, say from a book, is sequential or linear. There is a single linear sequence which defines the order in which the text should be read. The reader proceeds systematically from page one to the next page and so on until finished. The document is logically linear as well as physically linear. The author has assumed the responsibility to present the material in some particular logical fashion which guides the reader through a set of related material. In contrast, hypertext is nonsequential. "Hypertext presents

several different options to the readers, and the individual reader determines which of them to follow at the time of reading the text." [Nielsen, 1990a]

Hypertext is not really a new idea. Manual or traditional forms of hypertext have existed for centuries. Examples include the dictionary and the encyclopedia. These are forms of traditional text where the logical structure is separated from the physical structure. As one reads an article in an encyclopedia, explicit references to related information or sources are often made. The references "link" the article to additional related material. The reader then has the option to move directly to one of those explicit references to further explore the subject of interest. The reader is not constrained to read an entire encyclopedia in sequential order, and in fact it would not make sense to do so. Another example is the explicit references presented in this thesis which allow a reader to explore other reference material. Unfortunately for the reader it is not convenient to do so.

Hypertext is a set of information pieces connected with one another via pointers called links. Each of these information pieces or chunks are commonly called nodes. The entire set of interlinked information pieces forms an underlying network that is essentially a database of information nodes. (This database is commonly referred to as the hyperdocument or hypergraph.) The number of links in each node is not fixed, but is dependent on the content of the

node. Some nodes may be general in nature and have links to many other nodes. Other nodes only exist as a destination for a link from another node and may have no outgoing link at all.

A user "navigates" in a hypertext system. This is to distinguish the user from merely reading, as it is the user who decides which nodes to follow and in what order to pursue the information. Typically a user is able to simply point to a link and instantly move to the destination node. (For example with the click of a mouse.) Hypertext systems are thus nonlinear and encourage a nonsequential progression through the material.

On a more macro level there is no consistent or precise definition of what constitutes a true hypertext system. "...hypertext is an approach to information management in which data is stored in a network of nodes connected by links." [Smith and Weiss, 1988] According to Janet Fiderio;

> Hypertext, at its most basic level, is a DBMS that lets you connect screens of information using associative links. At its most sophisticated level, hypertext is a software environment for collaborative work, communication, and knowledge acquisition. Hypertext products mimic the brain's ability to store and retrieve information by referential links for quick and intuitive access. [Fiderio, 1988]

In an excellent overview of hypertext, Jeff Conklin

> ...focuses on machine-supported links (both within and between documents as the essential feature of hypertext systems and treats other aspects as extensions of this basic concept. It is this linking capability which allows a nonlinear organization of text. [Conklin, 1987]

Conklin further refines what a hypertext system is by eliminating what it is not. Although window systems have some of the feel of a hypertext system, they lack a single underlying database which is fundamental to any hypertext system. Many observers note the similarity of hypertext systems to DBMSs. "...DBMSs have links of various kinds (for example, relational and object-oriented links), but lack the single coherent interface to the database which is the hallmark of hypertext." [Conklin, 1987] Similarly Conklin disqualifies file systems, outline processors, and text formatting systems as lacking in the fundamental characteristics which define a true hypertext system.

Jakob Nielsen views a hypertext system as having a certain feel. It is this feel that allows users to move about the information space according to their own whims or needs. Part of this feel implies small cognitive overhead when using the computer.

> This means short response times so that the text is on the screen as soon as the user asks for it. Small overhead also requires low cognitive load when navigating, so that users do not have to spend their time wondering what the computer will do or how to get it to do what they want. [Nielsen, 1990a]

A common underlying thread in the definitions is that hypertext should employ a sophisticated notion of links and provide for a machine supported "feel" that allows the user to navigate through the underlying network or database according to their own desire.

### 2. Hypermedia

The term hypermedia has grown from the original concept of hypertext. This term has been adopted by many to stress the diversity of media used in the construction of nodes. Technology now allows for the nodes in a hypermedia system to consist of a large variety of media. Examples include text, graphics, audio, animation, pictures, video, or almost any other conceivable media. There is essentially little distinction between the two terms. Many authors consider the terms hypertext and hypermedia to be interchangeable. This is the convention that will be observed throughout the remainder of this thesis.

## B. A BRIEF HISTORY OF HYPERTEXT

The origins of present day hypertext systems can be traced to a 1945 article in the Atlantic Monthly entitled "As We May Think". In this article Vannevar Bush described a machine called the Memex as;

> A device in which an individual stores his books , records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory. [Bush, 1945]

Bush distinguished his concept from other forms of data storage and retrieval by using an associative structure that closely modeled his perceived structure of human memory.

> The human mind...operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts, in accordance

with come intricate web of trails carried by the cells of the brain. [Bush, 1945]

His vision of selection by association instead of indexing has guided hypertext designers for the past 45 years.

The term "hypertext" was first defined by Ted Nelson in 1965. His vision of Xanadu as a system that is a repository all of the world's literature represents one end of the spectrum of hypertext applications.

Some of the important milestones in the development of hypertext include the following [Smith and Weiss, 1988]:

- ZOG, a high-performance system developed at Carnegie-Mellon University and used aboard the USS Carl Vinson. ZOG is the predecessor of KMS, a modern day commercial system.
- Intermedia, and several earlier systems, developed by a research group at Brown University that traces its ancestry to Nelson.
- Notecards, the most ambitious system of the past decade, developed at Xerox PARC.
- Document Examiner, a beautifully engineered high-performance system by Symbolics that provides on-line access to their user documentation.
- Neptune, a system for computer assisted software engineering, developed at Tektronix.
- WE, an authoring system developed at the University of North Carolina that produces conventional paper as well as electronic documents and closely models human cognitive processes.

Several additional important events helped accelerate the interest in hypertext during the past few years. One of these was the introduction of Guide by OWL in 1986. This was the first available hypertext to run on personal computers. Another important event was Apple's introduction of HyperCard

in 1987. Apple's aggressive marketing of the system is credited with "...changing hypertext from an esoteric concept known to a few hundred people to a household staple of computing being used by millions." [Smith and Weiss, 1988] Lastly, the Hypertext '87 convention generated considerable momentum for this new technology when it brought together a broad spectrum of people in the first major conference devoted entirely to hypertext.

It is important to note from this brief history that, although hypertext enjoys a relatively rich history compared to some other developments in the computer field, the real momentum for research has only been over the past five years. There has been much ado about the hype in hypertext. Because it is a new and exciting field, some claims have been made prematurely that may be misleading concerning the capabilities of current systems.

> Often hypertext seems to be a solution in search of a problem; consequently, many hypertext designs appear to be driven by technology rather that by the needs of the users....The hypertext enthusiast focuses on its power to present users with myriad nodes of information from which to choose according to their desire for information. The skeptic, on the other hand, emphasizes the "confusing web of alternatives" that hides relevant information from the user, who will "pursue links of no relevance and arrive at relevant information without having viewed prerequisite, supporting information". Though we recognize the power of hypertext, our enthusiasm is tempered by the limitations of one part of the system - the user. [Herrstrom and Massey, 1989]

This passage illustrates there is indeed some "hype" in the technology of hypertext. The potential system developer

must have a thorough understanding of both the capabilities and limitations of the technology and combine this with the needs of the user when proposing hypertext as a solution to an existing problem. At this time hypertext is an immature technology with many fundamental problems in design, implementation and standards that are unresolved.

## C. HYPERTEXT ARCHITECTURE

Some of the basic components of a hypertext system have already been mentioned. The concepts of links, nodes, hyperdocuments, and navigation are all fundamental to the understanding of hypertext. This section will amplify the preceding information and provide the requisite background for understanding the design and authoring issues presented in Chapter IV.

### 1. Basic Architecture

There is no consistent fundamental architecture for hypertext systems reported in the literature. Most systems are comprised of an application layer/user interface and a basic underlying database structure. This has led to a collection of monolithic systems which have virtually no means of talking to each other. Examining hypertext systems built to date reveals a common underlying thread; "...virtually all systems have been insular, monolithic packages that demand the user disown his or her present computing environment to use the functions of hypertext and hypermedia." [Meyrowitz, 1989] Campbell and

Goodman have proposed a three level architecture that adds what they call a Hypertext Abstract Machine (HAM) that is sandwiched between the application layer and the database layer [Campbell and Goodman, 1988]. The HAM seems to be roughly analogous to a DBMS. The HAM is described as a general-purpose hypertext engine which can serve many types of hypertext systems. Some of the basic functions it performs include [Campbell and Goodman, 1989]:

- Create Operations - create new HAM objects.
- Delete Operations - mark objects as deleted but retain historical information.
- Destroy Operations - free all space required for an object.
- Change operations - modify data associated with an existing object.
- Get Operations - retrieve data from existing objects.
- Filter Operations - selectively retrieve information.
- Special Operations - include functions such as searchings for strings in nodes, merging contexts, and managing transactions.

The HAM is the best candidate thus far "...for standardization of import-export formats for hypertexts, since the database level has to be heavily machine dependent in its storage format and the user interface level is highly different from one hypertext system to the next." [Nielsen, 1990a]

Unfortunately, no current hypertext systems follow the model presented by Campbell and Goodman; "...they are a more

or less confused mix of features." [Nielsen, 1990a] However, the model presents a foundation for future standardization and is important in that respect.

Halasz describes architectural features of average second and current generation hypermedia systems. Second generation systems (e.g., NoteCards,Intermedia, KMS) differ from first generation systems (e.g., ZOG) primarily in the more advanced user interfaces allowed by the underlying workstation technology most of them were developed on. Table 3.1 is a reproduction of a table he uses to summarize the features of the "...average (and fictional) current generation hypermedia system." [Halasz, 1988]

### 2. Links

This section describes links in further detail and highlights their fundamental importance to any hypertext system. The question of how to build them will be reserved for the next chapter.

Links "...involve much more complicated theoretical and design issues than may at first appear." [DeRose, 1989] They are important to any hypertext system because they essentially define what the basic underlying structure (network) of the database will look like. The structure of this database has important implications for the system. Issues of navigation, data structures and display are all affected by this network.

TABLE 3.1 AVERAGE HYPERMEDIA SYSTEM FEATURES

| Feature | Description |
| --- | --- |
| Nodes: | Typed(text, graphics,...), implemented using a type hierarchy |
| Links: | Binary, bidirectional<br>Labeled but not typed<br>Anchors can be whole nodes or points/regions within the node |
| Overviews: | Browsers containing node/link diagrams of the network |
| Hierarchies: | Special support for hierarchical networks |
| User Interface: | Multiple windows; mouse/menu driven |
| Extensibility: | Programmer's interface |
| Search/query: | Slow, full-text string match |
| Distribution: | Single-user of multi-user central server with limited concurrency control |
| Versioning: | None |
| Storage: | Standard files or relational DBMS |

Usually a hypertext system requires multiple types of links that "...differ not only in purpose but in structure, function, and preferred means of implementation." [DeRose, 1989] There appear to be two fundamental types of links which Nielsen refers to as explicit and implicit [Nielsen, 1990a], and which DeRose calls extensional and intensional [DeRose,

1989]. DeRose further refines link types by breaking down these two categories into sub-categories and providing a basic taxonomy of links [DeRose, 1989]. Conklin also defines two basic types of links which he calls referential and organizational. Both of these types are classified by DeRose as extensional. Additionally, Conklin briefly refers to a third type of link called keyword link which is implicit in nature. DeRose also refers to a taxonomy of associative links proposed by Trigg which is fairly ambitious and covers a large range of needs. These associative links are a sub-category of relational links which are a sub-category of extensional links [DeRose, 1989]. Alschuler talks about sequential links and cross reference links which are called direct and indirect links.

> ..."indirect" link refers to cross-referenced nodes connected through an index or map. "Direct link" refers to links that jump directly from node to node. [Alschuler, 1989]

Specific hypertext systems often predefine link types used for that particular system. For example KMS defines only two types of links; tree items and annotation items.

> Tree items have the connotation of being linked to lower-level frames in a hierarchy, such as a Chapter of a book,...linked annotation items point to peripheral material, such as comments and cross-references. [Akscyn et al, 1988]

The subject of links seems rather confusing. There is no standardized terminology. For example both Nielsen and DeRose refer to implicit links. For DeRose these links are a

sub-category of intensional links. Nielsen seems to be using the term in the same context as DeRose's intensional links. The situation is further complicated by specific hypertext systems which, because they are designed for a particular purpose, use only a limited variety of link types. For example the KMS links referred to earlier are both explicit types of links. All of this makes for a rather fuzzy picture of link classification. The following paragraphs will attempt to clarify the picture somewhat.

As stated earlier there are two basic categories of links. The simplest classification seems to be Nielsen's categories of explicit and implicit. Most second generation hypertext systems consist entirely of explicit links. The explicit links give the underlying database its structure. "Most links are explicit in the sense that they have been defined by somebody as connecting the departure node with the destination node." [Nielsen, 1990a] The key is that explicit links must be stored individually. The following bullets briefly describe some types of explicit links but are by no means meant to be all inclusive.

- Referential Links - A non-hierarchical method of linking. Links are references that connect points or regions in the text. The source can logically be either a point or region of text. The link refers the reader to another node of information that is somehow related to the source node.

- Associative Links - A sub-category of referential links. These links tend to be unpredictable. Because associative links " attach arbitrary pieces of documents, they cannot be replaced by retrieval algorithms, or even by unilateral

creation on the part of an author [DeRose, 1989]. These links serve many purposes and are usually labeled by type.

- Annotational Links - A sub-category of referential links. Tend to originate from low level elements and represent connections from portions of a text to information about the text [DeRose, 1989].

- Organizational Links - These links implement hierarchical information. They connect parent to children and form a tree subgraph within the hypertext network. "They function mainly to represent super-ordinate/sub-ordinate relationships between documents." [DeRose, 1989]

Implicit links were apparently born of the limitations Halasz noted concerning navigation in a hypertext system;

> ...navigational access by itself is not sufficient. Effective access to information stored in a hypermedia network requires query-based access to complement navigation. [Halasz, 1988]

The need for this query-based access has caused a merging of hypertext and information retrieval technologies. Several papers presented at Hypertext '89 represent attempts at integrating information retrieval into hypertext systems. Coombs [1990] summarizes these papers noting benefits and deficiencies and proposes a full text search strategy which can be used to support automatic linking.

Implicit links follow from the structure and content of the documents they link, and are not stored explicitly in the system [DeRose, 1989]. "In other words the destination ...is defined by some function that finds the desired ends, rather than being a list of known ends." [DeRose, 1989] An example of implicit links is;

> ...the automatic glossary lookup possible in Intermedia. The InterLex server provides a link from any word in any

Intermedia document to the definition of that word in the dictionary, but it would obviously be ridiculous to have to store all these links explicitly. Only when the user requests the definition of a word does the system need to find the destination for the link. [Nielsen, 1990a]

Lastly it is important to highlight what links are used for. Hypertext provides machine support for tracing of references via linking [Conklin, 1987]. Links must allow the user to follow them easily and rapidly as he/she navigates through the hyperdocument. Conklin gives a list of link properties which may help the reader understand what links are supposed to do [Conklin, 1987]:

- They can connect a document reference to the document itself.
- They can connect a comment or annotation to the text about which it is written.
- They can provide organizational information (for instance, establish the relationship between two pieces of text or between a table of contents entry and its section).
- They can connect two successive pieces of text, or a piece of text and all of its immediate successors.
- They can connect entries in a table or figure to longer descriptions, or to other tables or figures.

Some additional uses might include:

- Connect a word to its meaning in a glossary
- Aggregate information into groups of nodes that represent common themes or "views" which are other than hierarchical.
- Allow a user to annotate a piece of information with a personal note or memo.
- Allow a user to query the database for a specific subject, name, or text string.

Appropriate links allow the user to navigate throughout the database in an unlimited number of meaningful ways. Machine supported linking is "...the essence of hypertext." [Conklin, 1987] Linking offers the advantage of letting the user know there is additional information on a subject; "...without hunting through an index or doing a search...." [Fersko-Weiss, 1991] This access is transparent; a user does not need to know what file or application the information resides in.

### 3. Nodes

Nodes are the other basic building block of all hypertext systems. As in links there is no standardization between systems as to what constitutes a node or how these nodes should be presented to the user. Although node seems to be the most generally accepted term, different systems call them by different names. For example in KMS, nodes are referred to as frames and in NoteCards the nodes are (not surprisingly) called notecards. Whatever the name the concept is the same. Nodes represent a partitioning of the document according to some common scheme. How this is done is significant to the user because the "...structure and design are important in determining just how readable a document is." [Fersko-Weiss, 1991] Every system defines nodes in a different way. Nodes may be almost any size, may be fixed or scrolled when presented to the user, may use different data

models, and may contain almost anything. The composition of nodes is at the discretion of the author of the document and is therefore as diverse as individual authors are. Many documents do not lend themselves to simple partitioning because of the interleaving of ideas and themes throughout the text. The issue of how to break an existing text into discrete blocks of information is a very complex design issue that requires individual judgements.

Generally hypertext systems have nodes which express a single idea or concept. "Hypertext invites the writer to modularize ideas into units in a way that allows (1) and individual idea to be referenced elsewhere, and (2) alternative successors of a unit to be offered to the reader...." [Conklin, 1987] While most systems provide fixed information in the nodes, with some "...computational hypertext systems like KMS and HyperCard...it is possible to have computed nodes generated for the reader by the system." [Nielsen, 1990]

Nodes can be typed or untyped. "An untyped node is a box for information." [Fiderio, 1988] KMS provides for only one type of node(frame) and variety is provided by individual items within the frame. There is no restriction for what can be put into a KMS frame. A typed node has a label which allows the user to determine the style of information contained in the node. Types can also be used to provide a particular structure or define specialized operations. For example, gIBIS

is a system designed for systems analysis of complex problems which uses three kinds of typed nodes. Individuals can use issue nodes to describe an issue to be discussed; position nodes to describe an assertion that resolves an issue; and argument nodes which contain your objection or support for a position.

Generally nodes are unstructured. However applications are being developed that use a structured node or semistructured node type. These semistructured nodes may provide a template for node contents to help the user ensure completeness and assist the computer in processing [Conklin, 1987]. An example of the semistructured nodes are the issue, position, and argument nodes of the gIBIS system noted earlier. Creation of a node is done via a template to fill in the node's structured fields. When complete the node is "...parsed and stored and the browser and index windows are updated to include it." [Begeman and Conklin, 1988] Another example is [Jordan et al, 1989]:

> Instructional designers have found Template cards particularly useful since their task involves recording textual information with a standard format. Template cards accelerate the creation of networks by allowing the user to specify in advance text common across cards, removing the need for redundant retyping and reformatting. [Jordan et al, 1989]

The requirement for composite nodes was stressed by Halasz as one of the seven issues for the next generation of hypermedia systems. Composite nodes are aggregations of related nodes that form a higher order entity. They are "...a

way of representing and dealing with groups of nodes and links as unique entities separate from their components." [Halasz, 1988] A composite node facility allows a group of nodes to be manipulated as a single node [Conklin, 1987]. The need for some type of composite node seems clear. Hypertext documents are highly modularized with each node often only representing a single concept. There are times when it is easy to imagine that a user would like access to an entire chapter at once rather than having to refer to the individual nodes in some sort of linear fashion.

Conklin also talks about the "...idea of building a directed graph of informal textual elements...similar to the AI concept of semantic networks." [Conklin, 1987] In this scheme concepts are represented by nodes, and relationships between concepts are links between the nodes. It seems that an analogous hypertext network could be built, with ideas and concepts represented in nodes and semantic interdependencies between them represented as links. Semantic network work suggests some extensions to hypertext where typed and semistructured nodes can be used to effectively implement inheritance hierarchies of node and link types. [Conklin, 1987]

## D. APPLICATIONS

### 1. General Categories

The domain of hypertext applications is exceedingly broad. Some liken it to the diversity represented in printed material [Nielsen, 1990]. Different kinds of applications require different kinds of hypertext support and this is reflected in the great diversity and functionality of current hypertext systems. Several authors have referred to general categories of hypertext applications. Four broad application areas include [Conklin, 1987]:

- Macro literary systems - these systems support large on-line libraries with machine supported interdocument links. Nelson's Xanadu is an example of this type of system.
- Problem exploration tools - these systems provide tools to support early unstructured thinking on a problem when many disconnected ideas come to mind. The gIBIS system referred to earlier in the paper is an example of this category.
- Browsing systems - these are similar to the macro literary systems but on a smaller scale where ease of use is critical. ZOG, KMS and Hyperties are examples of this class.
- General hypertext technology - these are general purpose systems for experimentation with a range of applications such as reading, writing or collaboration. NoteCards and Intermedia are good examples of this group.

Halasz applies a different perspective when classifying hypermedia systems. According to him, the diversity of hypertext systems can be partitioned along "...three fundamental dimensions: scope, browsing vs authoring, and target task domain." [Halasz, 1988] These are explained below [Halasz, 1988];

- Scope - The scope of hypertext systems range from a proposal as the mechanism for storing all the world's literature to projects as small as a tool for individuals and small groups engaged in authoring and idea processing. This extreme variation in scale implies large differences in design of everything from storage mechanisms to user interfaces.

- Browsing vs Authoring - Systems designed primarily for browsing are created by a few authors to provide a network for exploration by a large number of relatively casual users. Tools for creating or modifying the network are less developed. Instructional delivery systems are good examples. Authoring systems serve as an information structure which users create and modify as part of their regular tasks. Examples include systems for document authoring and software development.

- Target task domain - Many systems were designed to support a specific task. Even other more generic systems were designed to support a target task domain and thus differ substantially in the features and capabilities emphasized during design. For example Intermedia and Neptune are both general purpose systems but Neptune was designed to support software engineering and Intermedia was designed for multiuser interactive educational applications. Neptune emphasizes versioning and node/link attributes while Intermedia emphasizes interactive displays and annotation facilities.

These are particularly useful dimensions for examining the design of a hypertext system. The browsing versus authoring issue seems particularly important because most applications generally fall into one of these two categories. These categories have explicit differences in fundamental design issues such as linking and node construction. For example, ease of navigation and information retrieval are very important in a browsing system. Because certain navigation modes can only be supported by certain network structures, (see Chapter IV for more detail about navigation) it is

ultimately the user needs(browsing) or task domain the system was designed for that affect the database structure.

This issue of browsing versus authoring also seems to divide developers into two different camps with different views about what constitutes true hypertext. Those that support authoring "...contend that unless the user can make bi-directional links--or chart new paths through the information space--the system is not truly hypertext." [Carlson, 1989] These designers support an underlying web structure to facilitate learning, creativity, and collaboration. These people view applications which use hypertext as an information management system for storing and accessing huge bodies of text as little more than DBMS technology. Probably the best approach is to view both applications as opposite ends of a broad spectrum of applications.

### 2. Specific Application Overview

There are a number of considerations when deciding to apply hypertext to a specific problem. Certainly "... not everything that can be put online should be put online; people prefer some ties to forms and practices with which they are familiar." [Grice, 1989] "Just because hypertext is used for an application does not ensure that a user's needs are served." [Shneiderman, 1989] Three golden rules of hypertext have been proposed which should be used to identify key

attributes of hypertext projects. These rules are [Shneiderman, 1989]:

- there is a large body of information organized into numerous fragments,
- the fragments relate to each other, and
- the user needs only a small fraction at any time.

There are countless specific applications outlined in the literature. Nielsen [1990a] devotes an entire chapter in his book to talk about them. These applications fall into general categories which include computer applications such as online documentation, business applications such as online repair and other manuals, intellectual applications like idea organization and brainstorm support, educational applications and entertainment applications. Fersko-Weiss[1991] notes several categories of commercial applications which include application generators, help systems, personal information managers, expert systems tools and authoring programs. Applications specific to the Department of Defense are addressed in Chapter V.

## E. PROBLEMS IN IMPLEMENTING HYPERTEXT

Disadvantages of hypertext have been well documented. These deficiencies come in two categories: "...problems with the current implementations and problems that seem to be endemic to hypertext." [Conklin, 1987] Two problems from the

latter category which need to be addressed are "...disorientation and cognitive overhead." [Conklin, 1987]

### 1. "Lost in Hyperspace"

One of the best ways to describe this problem is to again contrast hypertext systems with something familiar such as a book. When reading a new bock few people become disoriented or feel uncomfortable using it.

> We know how to leaf through it (forward or backwards), how to place our fingers between the pages to mark places we may want to return to, how to read it completely, and how to put it down when we no longer want to look at it. [Grice, 1989]

If it is a reference book we are generally comfortable searching it for whatever information we require. We understand the hierarchical organization of the book into chapters, and possibly sections and subsections. We know how to search the index in multiple ways to seek references for the desired material and we know there is a table of contents that might be useful. Additionally the author has imposed some form of logical structure on the material which is simple to follow. In short we are used to printed material and are comfortable using it.

Hypertext systems present us with an environment we are not used to coping with. Users might not know where the relevant information is, but they can only search in two directions; forwards or backwards. Disorientation is uncommon. On the other hand, hypertext systems require the user to

determine the order of the information he/she will pursue. Lines of pursuit can follow many trails. There are many more dimensions where the user can travel, and now orientation becomes an issue. Most of the visual cues which we are familiar with in printed material are absent. One can readily determine how long a book is or for that matter how much material is left in a chapter we are reading. Information like this which we unconsciously use to decide how much more to read or where to go to next in printed material is likely missing in a hypertext system. It is not difficult to imagine how easy it is to become disoriented when a user begins to pursue a single line of thought and then branches off along other associative lines within the network. How often has each of us "lost our train of thought"? This problem is especially compounded if there is no visible form of structure to the hyperdocument. In order for the hypertext system to be beneficial, the user must be able to "...know (1) where you are in the network and (2) how to get to some other place that you know (or think) exists in the network." [Conklin, 1987] If a user cannot find the information he/she needs, the system has little value.

The real problem with this disorientation is successful information retrieval. When a user is exploring the database for the sake of exploration alone; when the goals are learning or creativity; then the disorientation problem may not be a problem but rather a benefit. Under these

circumstances, searching a database which has no structure imposed on it may be very exciting and rewarding. However if the search is motivated to retrieve specific information to solve a specific problem and there are time constraints imposed on the task, the issue of navigation becomes important. Each circumstance reflects the issue of purpose the hypertext system is to serve.

### 2. Information Retrieval

The problem of information retrieval manifests itself in many ways. An example cited earlier referred to users following links of no relevance and arriving at relevant information without having seen prerequisite supporting information. This is a function of having numerous paths through the hyperdocument with no guidance to the user in terms of logical structure. The user may arrive at a particular node from a number of paths and have little idea of the context of the node with respect to the rest of the database structure. Worse yet, if there is no mechanism for the user to build paths or trace his/her path to a node, relocating the node at a later time might be almost impossible.

Another problem of information retrieval is recognizing useful or relevant information. "Since the majority of users do not possess the specific knowledge of subject area specialists,...they find it difficult to separate

irrelevant information from relevant information for the successful completion of their tasks." [Rubens, 1989] This that the user may bypass information later recognized as useful. The problem is to relocate the information which was previously found. Even if the information was found just a few nodes ago this may be difficult. Nielsen documented problems users experienced navigating a hypertext document; Some 44% agreed with the statement "When reading the report, I was often confused about how to get back to where I came from." [Nielsen, 1990] The issues of information retrieval and system navigation seem highly integrated and pose serious design issues.

The large volume of data represented in a hypertext database may be another problem. Often, more information is implied as better, however;

> In reality, more information simply requires the user to consider and process more information that may be indiscriminable. The path to a useful response may, in fact, get even more obscure in such data pools.[Rubens, 1989]

This passage reflects the problem of locating and recognizing relevant information. It also alludes to the other problem endemic to hypertext; high cognitive overhead.

### 3. Cognitive Overhead

The problem of high cognitive overhead implicit in hypertext systems is well documented. It exists for both builders or authors of systems, and users or readers. For

authors, "...it is difficult to become accustomed to the additional mental overhead required to create, name, and keep track of links." [Conklin, 1987] For the readers of hypertext documents the overhead exists in the "...large number of choices about which links to follow and which to leave alone." [Conklin, 1987] Unlike a book where the author guides the reader through the logical structure, the hypertext system places this burden on the reader. The design of the system may emphasize this problem. If the destination of the links is not obvious or there is a slow response time when traversing the links, overhead is added for the user when trying to decide which path to pursue.

### 4. The Problem of Scope

Many of the problems noted with hypertext systems have occurred on relatively small systems with limited scope. The magnitude of complexity of the above issues is relatively untested in very large hypertext systems. The nature of the hypertext document database breaks the document into small pieces or nodes. One must imagine what 300,000 pages of F-18 documentation might look like assembled into a hyperdocument. It is also easy to imagine the volumes of information that might remain hidden, buried in the maze of nodes and links. Little documentation exists concerning really large hypertext databases. The many as yet unimagined problems sure to surface when implementing truly huge hypertext databases represent

problems of the future. The risk involved must at least be recognized before embarking on a program on a scale many times greater than those attempted thus far.

# IV. BUILDING A HYPERTEXT SYSTEM

Building a hypertext system is a complex process. This chapter will amplify previous information and discuss some current design issues.

## A. THE INFLUENCE OF THE USER AND HYPERTEXT USEABILITY

### 1. User Influence

As noted in Chapter III, hypertext often seems to be a solution in search of a problem. It is thus important for the prospective developer to recognize and understand the actual needs of the user before blindly applying the technology. "Hypermedia may be an excellent solution to some communication problems; it may be an overpowering solution to others." [Grice, 1989]

The developer must find out both what the users want and what the users need. This is not easy to do but is a necessary first step. "User task demands vary widely, so it should not be surprising that a variety of hypertext models are needed to support a variety of tasks...." [Perlman, 1989] As discussed in Chapter III, the users tasks fundamentally drive a great deal in a hypertext system design from the underlying database structure to the user interface.

### 2. Hypertext Useability

The useability of a hypertext system is a primary design concern which is driven by the end user. Grice [1989] notes three characteristics of online information which must be present if people are going to use the system. The system must be useful, simple to use and attractive. Grice also identifies a number of factors which affect whether people will use the system. These factors include [Grice, 1989]:

- Familiarity
- Support of user's tasks
- Ease of Learning
- Navigability
- Ease of modification
- Appearance

The point of familiarity is one especially worth noting. This point includes familiarity with the subject matter, presentation and conventions used [Grice, 1989]. The movement from printed material to online information represents a major change for many people. This problem may be acute in DOD where we are so heavily dependent on documents for the performance of our daily work. People are naturally resistant to change, so anything that can be done to ease the transition is helpful. "Several systems have advocated a book metaphor so that the online version of information closely matches printed formats, even if there is no printed version." [Perlman, 1989] Kellet [1989] promotes the use of existing

conventions such as tables of contents, indexes, glossaries, references, etc, to give electronically stored Navy manuals a familiar shape and structure while still allowing the benefits of hypertext capability.

The overall useability of a hypertext system is determined by a "...combination of the usability of the underlying hypertext system engine (i.e. the basic presentation and navigation support available) and the usability of the contents and structure of the hypertext information base, and by how well these two elements fit together." [Nielsen, 1990a] Five components of useability include [Nielsen, 1990a]:

- Easy to learn: Basic commands and navigation tools are easily understood and used to locate desired information. Also the contents of each node are easy to read and understand.

- Efficient to use: Users are able to quickly find desired information or easily ascertain that the information does not exist in the database. When users arrive at a node, they readily orient themselves and understand the relationship of the node to the point of departure.

- Easy to remember: After a period of non-use, users have no trouble remembering how to navigate through the system to locate desired information.

- Few errors: Users seldom follow links to someplace they did not really want to go. Few links exist that go nowhere or to erroneous places. Information in the nodes is correct.

- Pleasant to use: Users do not become frustrated using the system and prefer the system to existing alternative procedures. Users are in control and do not feel constrained by the system.

While most of these factors appear to be a user interface issue, in fact the underlying structure of the database also determines much of the "look and feel" of the system.

### B. CONSTRUCTION OF THE DATABASE

The structure of the database consists of two parts: the underlying data model is concerned with the organization and layout of the individual nodes, while the organization of the entire set of nodes affects the navigation options available to the user and largely determines the ease with which relevant information is located.

#### 1. The Data Model

Much of the useability of a hypertext system is a function of how the nodes are constructed. "One of the most pressing problems of human-computer interaction is the ever growing complexity of software systems." [Akscyn et al, 1988] They attribute this complexity to the complexity of the underlying data models.

> If there is one central lesson from our experience, it is the fundamental importance of a system's data model. Our experience with ZOG and KMS has convinced us that the data model underlying an interactive system strongly determines its 'look and feel'. [Akscyn et al, 1988]

They note that it is the properties of the frames (nodes) in KMS, for example their fixed size, spatial nature, how links are represented, etc., which "...contribute to the global nature of the system." [Akscyn et al, 1988] There are

many design issues concerned with the data model. Some of these issues which include determining [Akscyn et al, 1988]:

- What size a node should be.
- What types of nodes there should be.
- What types of data objects should be used as the source for a link.
- What types of data objects should serve as the destination for a link.
- What types of links there should be.
- What is the structure of a link.
- How nodes can be aggregated into larger structures.
- How versioning is to be supported.

### 2. User Interface Issues

There are a number of user interface issues which are closely related to the design of the data model. Style of the user interface, for example menus, direct manipulation via mouse, etc, must be selected. The designer must also determine how nodes should be displayed. There are various styles which include (1) presenting nodes in separate windows, using multiple windows of different sizes, (2) linear display where each node is expanded in place, and (3) full screen or split screen presentation [Akscyn et al, 1988]. The developer must also determine how to display link sources and link destinations. Examples of this are use of text highlighting, embedded icons, or whole text items. System response time is an important issue. Some believe that "...fast system response

to selecting a link is the most important parameter of a hypermedia system." [Akscyn et al, 1989] System support for browsing, use of graphical views, and information search and retrieval are other user interface issues which will be covered in more detail later.

### 3. Database Organization

An important point to remember is that hypertext systems "...must match the information available to the readers' need to access that information." [Brockmann et al, 1989] Although hypertext offers new ways of accessing information, simply adding links to a document does not make it more useful. Some form of useful organization is still required;

> Since the developer, in all but a few hypermedia systems, creates the links, users still must trust the rational approach, as well as the mindset, of the developer. To avoid the all-to-frequent dilemma of GIGO (garbage in, garbage out), some intelligent structuring must occur early in the development cycle. [Rubens, 1989]

Hypertext systems support numerous database organizational structures. These tend to fall into categories which generally support whichever use the system was designed for. Database organization affects the navigation and information retrieval options available to the reader. Two important categories are hierarchy and web. The web structure is often used for systems which support database exploration, where the primary goals are learning and creativity. A hierarchy is commonly used for task driven systems where the

user is searching for a specific piece of information to support some work related activity.

"The power and risk of hypertext can be seen by looking at the expressive power it adds to the organization of documents." [Brockmann et al, 1989] Figure 4.1 outlines four common structures for organizing documents [Brockmann et al, 1989].

The sequence is the simplest of the structures and is the scheme used for most paper documents. It is highly predictable in that the reader only has two choices for exploration, forward or backward. The expressive power is very limited.

The grid is a familiar form of structure which allows the addition of another logical dimension without a great loss of predictability. Brockmann et al [1989], illustrate this form with an example of a word-processing reference manual. The columns of the grid represent the different commands for the system. Each command has a set of common characteristics or descriptors such as syntax, description, notes, and examples which make up the rows. This structure allows the reader the options of reading the columns to find out everything about a specific command, or across the rows to compare the same feature of various commands.

The tree or hierarchy is also a very familiar form of structure which is used for classification and management. Hierarchies can be somewhat confining (less expressive) in

that the author "...must anticipate the most important criteria for later access to the information." [Conklin, 1987] This may be extremely difficult because different hierarchies can be constructed from the same data base depending on the frame or context from which the builder is working. A classic example is the classification scheme used for four objects: a watermelon, an orange, a baseball, and a football. There are obviously multiple schemes possible depending on whether shape, origin (natural or artificial), size, or some other characteristic is used as the starting point [Van Dyke Parunak, 1989]. "As Vannevar Bush said at mid-century, the organization of information into hierarchies buries important information in 'the mass of the inconsequential'." [Coombs, 1990] Hierarchies may be somewhat unpredictable in that when moving down the tree multiple choices and paths are offered which make it more difficult (than a sequence or grid) to maintain orientation in the database. This is not to say that orientation in a hierarchical system is difficult. For example in KMS ;

> The resulting hierarchical "skeleton" in the database helps users build a coherent mental model of the database. They can remain oriented when navigating because they can always see whether they are selecting a hierarchical link or a cross-reference... [Akscyn et al, 1988]

Hierarchies are attractive because "...abstraction is a fundamental cognitive process, and hierarchical structures are the most natural structures for organizing levels of abstraction." [Conklin, 1987] Hierarchical organizations

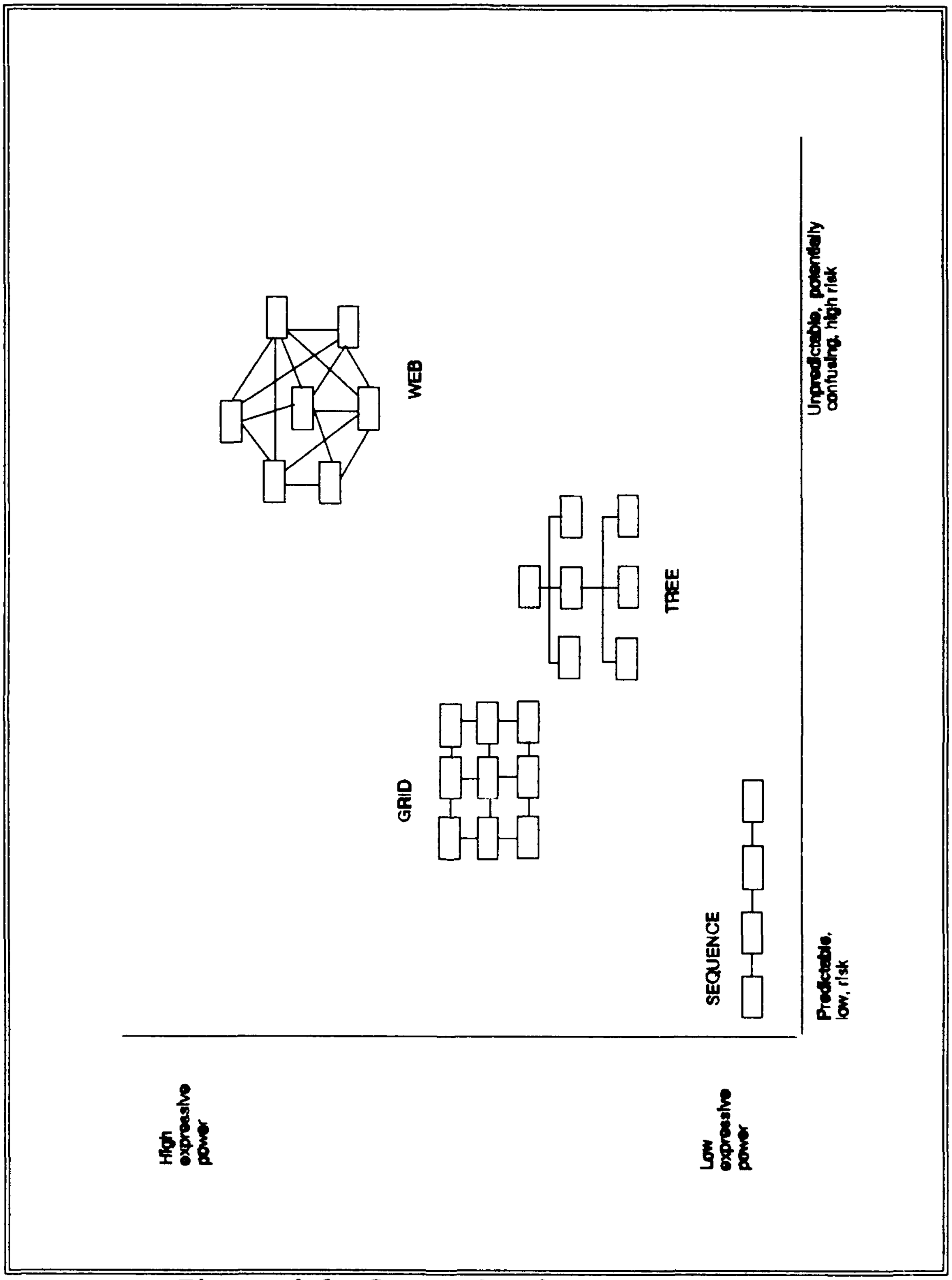


Figure 4.1 Common Database Structures

"...have proven so compelling that we can expect them to be dominant well into the 21st century, and no one is well served by research that fails to make the most of the structures that people typically create." [Coombs, 1990]

"The web structure...is the ultimate in expressive power." [Brockmann et al, 1989] In this structure anything can be connected to anything else through association. They are not constricted to a specific set of rules or structure. "The power of the web structure is that with it the designer can construct other simpler structures, as well as special ad hoc structures, such as cycles, stars, and diamonds...." [Brockmann et al, 1989] The problem with the web is that it lacks the structure which may guide a reader to desired information. A web may produce "...sprawling unmanageable systems having little overall coherence." [Brockmann et al, 1989]

The structure of the database is a function of the links used to join the nodes. The topology of links that join nodes together is "...one important criterion for classifying hypermedia...." [Van Dyke Parunak, 1989] Reviewing the earlier classification of links it reveals that a hierarchical structure is supported primarily by organizational links while a web structure is supported through referential links. In general, useful hypertext applications require multiple types of links, both of the explicit and the implicit variety. Conklin [1987] refers to work by Frank Halasz which indicates

"...that hierarchical structure is very important in organizing a hypertext network, and that referential links are important but less common." [Conklin, 1987]

The explicit links in a hypertext system define its structure. This structure in turn defines navigation strategies available to the user. These strategies directly impact on the useability of the system which, for the user, is a prime component for determining the usefulness of the system. "Usefulness is the issue of whether the system can be used to achieve some desired goal." [Nielsen, 1990a] Lastly, the issue of usefulness is an essential sub-component (from a users perspective maybe the only important component) of the practical acceptability of a computer system. If a system is not practically acceptable it has little value. The next step in following this chain is to look at the navigation and information retrieval options available to a system user.

## C. NAVIGATION/INFORMATION RETRIEVAL

Both navigation and information retrieval are means for hypertext system users to locate information. "When users move around a large information space as much as they do in hypertext, there is a real risk that they may become disoriented or have trouble finding the information they need." [Nielsen, 1990a] Some of these problems were documented earlier. Indeed, disorientation is one of the endemic problems of hypertext which has attracted much research attention.

Hypertext systems will have little value if this problem can not be solved. Navigation (navigation is also often referred to as browsing) tools are designed to help orient the user and minimize the "lost in hyperspace" feeling. This expedites the location of relevant information. However, navigation tools alone have been demonstrated to be insufficient for information location. "...the browsing paradigm offered by most hypertext systems does not meet the needs of readers who are unfamiliar with the material being presented." [Zellweger, 1989] Thus alternate solutions to this problem are being researched.

### 1. Navigation Solutions

There are a number of solutions to the navigation problem which all help in some respect. However, no single solution is a cure all for the problems associated with navigation. One of the more common solutions to this problem is the graphical browser.

> Browsers rely on the extremely highly developed visuospatial processing of the human visual system, by placing nodes and links in a two or three dimensional space, providing them with properties useful in visual differentiation (color, size, shape, texture) and maintaining certain similarities to our physical environment...designers are able to create quite viable virtual spatial environments. Users orient themselves by visual cues.... [Conklin, 1987]

An example of a graphical browser is an overview map. The very large size of some information spaces often requires these maps to show various levels of detail. Examples are

global views which show the information space on a high level and local views which show a much greater level of detail for the current location in the information space.

A similar alternative is a fisheye view which shows information in the same space with various levels of detail. More detail is shown in the center of the display screen for the information space close to the users current location. Detail becomes progressively less as the viewer moves from the center of the screen to the perimeter. This type of view is more appropriate for hierarchical structures because it requires that it be possible to estimate the distance between a given location and a users current location, and that information can be displayed at various levels of detail [Nielsen, 1990a].

One of the most important navigation mechanisms for any system is a backtrack feature which allows users to retrace their steps until they are in familiar territory. This includes the ability to retrace steps all the way back to the introductory node. Also, "...backtrack facilities need to be simple and consistent, so that users can always rely on them as a lifeline to get out of trouble." [Nielsen, 1990b]

There are several forms of historical mechanisms which assist navigation. "For example some systems have history lists...to allow users direct access to any previously visited node." [Nielsen, 1990a] This aid ensures the user will be able to relocate previously found information which might

otherwise be difficult in a large information space. Other systems allow users to place bookmarks they later might want to return to. The advantage is that the bookmark list will be smaller that a historical list and thus be more manageable and contain a greater percentage of relevant information. The drawback is that the user must recognize information as important when first seeing the node. This is often not the case, which may mean the bookmark list does not contain all of the relevant information. [Nielsen, 1990a]

Another form of historical navigation aid is the use of time stamps. This type of system tells a user how long it has been since the last time the user viewed the information, or if it's the first time the user has seen the node. "One will view information in different ways depending on whether one has never seen it before or one has seen it a short or a long time ago." [Nielsen, 1990b]

Use of "landmark" nodes may also help users orient themselves within the database. These nodes would be especially prominent nodes within the information space. Candidate nodes for landmark status can be calculated based on a measure of connectivity. This is a means of identifying nodes that are either referenced by many other nodes, or contain links to many other nodes. Prominent landmark nodes in an overview diagram of a web structure might be especially useful to assist the reader with orientation and to suggest good entry or exit points.

Another means of simplifying the overall picture of an underlying information space is the use of link inheritance. This can be used to group together nodes into related clusters and link the clusters on the overview diagram. Here again this might be more useful for an underlying web structure where there is no apparent form of organization to help orient the user. As a matter of fact some maintain that "...views are limited in value, except perhaps for large, essentially non-hierarchical structures." [Akscyn et al, 1988]

"An important component of the information conveyed by an author to a reader in a traditional setting is the order in which the material appears." [Zellweger, 19 ] In many hypertext systems readers fail to understand the information presented because they view it in the wrong order, or entirely miss important pieces of prerequisite information. One proposed solution for this is to remove the requirement for the user to direct the navigation by having the system provide paths or guided tours through the information space. "Users are less likely to feel disoriented or lost when they are following a pre-defined path rather than browsing freely, and their cognitive overhead is reduced because the path either makes or narrows their choices." [Zellweger, 1990] These paths are very beneficial to the user who needs assistance finding relevant information in the proper sequence. However, this solution is a linear presentation formulated by the original author, and may be somewhat restrictive for the user.

These solutions are intended to augment other navigation efforts not replace them.

The navigation tools discussed thus far, are designed to help the user move about the overall information space giving them a sense of context and location in the overall structure. There is an additional navigation problem associated with individual nodes which is referred to as "...context in the small". [Nielsen, 1990b]

> When the computer display is as small as in our system, users can only see a very small part of the information at any one time. This means that they can very easily lose track of how the text they are currently reading is related to the immediately preceding or following text. [Nielsen, 1990b]

Because nodes can contain anything and be any size, a user needs some indication of what is contained within a node. This is especially true if a node is not presented in its entirety on the display screen. If a node contains multiple pages, the reader needs to understand where they are in the node, the relative size of the node, how much information precedes their location, and how much information follows their location. Techniques for dealing with this problem include increasing the size of the screen or display window, limiting the size of a node to a single screen and providing some visual indication such as a gage or scroll bar which indicates relative size of the node and position within the node.

### 2. Information Retrieval

There are a number of factors which, combined, make it "...practically impossible to abolish the disorientation problem with a browser alone." [Conklin, 1987] The problem becomes more acute as the size of the underlying database grows. The difficulty locating relevant information in a database containing thousands of nodes and thousands of links is not hard to imagine no matter how the database is structured or how many navigation tools are present. Some form of information retrieval technique is required to assist in the location of relevant information.

Halasz [1988] maintains there are two general categories of "query/search mechanisms" required for a hypertext system; content search and structure search. Content search ignores the structure of a network and searches each node and link for a match to a user query. An example of content search is full text search which finds matches to specific text strings or keywords input by a user query. On the other hand structure search allows a user to query the system for a diagram of all subnetworks that match a specific pattern.

> For example, the following is a simple structure query: all subnetworks containing two nodes connected by a supports link, where the destination node contains the word "hypertext". [Halasz, 1988]

A somewhat more sophisticated notion of content based search is the notion of inference for retrieval. To move

beyond simple specific string matching retrieval mechanisms, "...determining that the query and a node are related will require an inference mechanism." [Croft and Turtle, 1989]

> In short, one has a variety of reasons to infer that a particular node, or document, is about a particular topic. Crucially, if node 1 is about concept C and node 2 is linked to node 1, then there is some reason to believe that node 3 is also about concept C. [Coombs, 1990]

Examples of inference include statistically derived nearest neighbors, direct citations, structural hierarchies, and manual links. These inference techniques allow retrieval of nodes that might be otherwise missed by a simple text string or keyword search.

The use of intelligent access mechanisms for large complex databases is another form of information retrieval. The goal of these systems is "...to help the user select a path through the textbase that is tailored for a particular application or purpose." [Carlson, 1989] These types of interfaces assist a user in refining their query to improve the relevance of information retrieved.

Query mechanisms can be used for more than simply locating specific information. They can be also used as a filtering mechanism for navigation tools such as overview diagrams. For example, a query could be used to filter the database to display only those nodes relevant to the query. This would simplify user navigation through the space of relevant information and improve retrieval.

There are many more information retrieval mechanisms in existence and being researched than can be covered here. The point of this discussion is to emphasize the variety of mechanisms, their multiple uses, and their importance to most hypertext systems. They become especially important as the size of the database grows.

## D. AUTHORING HYPERTEXT

Authoring a hypertext system falls two into broad categories. Hypertext systems may be authored completely from scratch, or they may be a conversion of existing text into a hypertext environment. While a better hypertext product may be produced by writing nodes from scratch, the huge amount of already existing printed material suggests it may be even more important to devise methods to produce the latter form of hypertext system [Nielsen, 1990a]. This seems especially true for the large number of potential Department of Defense hypertext applications. (See Chapter V)

When converting existing text to hypertext the author must first ensure that the document to be converted is appropriate for a hypertext environment and secondly take great care to ensure the design of the system is good. If instead the opposite is true, then hypertext may "...lead to hyperchaos." [Shneiderman, 1989]

"Hypertext basically destroys the authority of the author to determine how readers should be introduced to a topic."

[Nielsen, 1990a] This is not to say that the author is relieved of any responsibility for structure of the hypertext document. While readers may be free to explore the database in any fashion they choose, the author "...still has the responsibility to provide certain priorities for the readers and to point them in relevant directions." [Nielsen, 1990a] Breaking a document into small nodes and links does not guarantee that the resulting product will be effective.

> Successful hypertext, just as any successful writing project, depends on good design of the contents...The hypertext author...must take great care to produce excellence. The designer who assumes that it is safe to throw everything into the hypertext network and let the reader sort it out will be surprised by the negative reactions. [Shneiderman, 1989]

One of the difficulties noted with hypertext authoring is the "...problem of premature organization." [Halasz, 1988] Many hypertext systems require the author to structure the information at too early a stage. This presents organizational problems because;

> A user in the very early stages of working with a particular set of information may not sufficiently understand the content and structure of that information. Knowledge about the critical dimensions of the idea space, the characteristics which distinguish one idea from another, and appropriate naming schemes develops over time as the user becomes familiar with the information. [Halasz, 1988]

The complexity of authoring hypertext systems is also caused by the fact that hypertext systems borrow "...characteristics from a variety of other media." [Marshall and Irish, 1989] For example, hypertext authors are still

required to provide context and create coherence as in traditional forms of writing, and they are additionally required to resolve problems of screen layout and related activities which require the skills of a graphics designer. Hypertext's great potential for interactivity suggests questions of design from areas such as Computer Aided Instruction, and hypertext traversal has qualities suggesting characteristics and associated difficulties of film-making or animation [Marshall and Irish, 1989]. All of this implies that a number of skills are required of hypertext authors if a truly useable and effective system is to be built. This also implies that teams consisting of a variety of skilled personnel are better suited for building complex hypertext systems than individual authors.

### 1. Making Nodes and Links

Basic hypertext authoring boils down to construction of nodes and links. The author is required to make value judgements concerning "...what to include in the database, what type of links to create, and how to organize topics." [Fiderio, 1988] While no set rules exist, some basic guidelines seem to be consistent.

The information should be organized into small "chunks" with each chunk representing a single idea or concept. Focusing on a single topic allows a user to more easily recognize a node in overview diagrams or history lists.

It also conforms better to the medium. Computer screens tend to be small and hard to read, so minimizing the amount of on-screen text is less burdensome for the system user. The nature of hypertext supports this because subsidiary topics, definitions, etc. can be placed into other nodes which allow users to access them only if desired [Nielsen, 1990a]. This also allows the presented material to conform better to individual user preferences and needs. A novice or inexperienced reader has all the detail available that an expert might find distracting.

The general strategy for linking is to be judicial with the use of links. There should be a variety of links but the author should take care to avoid trying to link everything. Too few links may indicate that the use of hypertext may be inappropriate, while too many links creates a great deal of cognitive overhead for the users and may overwhelm them [Nielsen, 1990a]. The types of links used must be decided on.

The decision of what to link can be extraordinarily subjective. Liora Alschuler studied three different hypertext systems constructed from the same six articles which were each presented at the Hypertext '87 conference [Alschuler, 1989]. While there were differences between the three platforms used to build each system, each was constructed using what she refers to as "hand-crafted" linking. This means there was no random, intelligent or automated linking, but instead all

links were a result of the individual determination of the authors. The study was revealing, illustrating that;

> Even in so limited a task as connecting six related magazine articles, there are vast differences in the way information can be structured. The difference between the programs is not merely one of speed, efficiency, elegance or the care with which they are presented...but a qualitative difference due, at least in part, to inconsistent adherence to document structure and to the subjective nature of their linking systems. [Alschuler, 1989]

One of the key points of the study is the tremendous difficulty with the linking process, and just how hard it is to convey to the user what the contents of the system are and how to navigate in the system to find relevant information.

The problems with "hand-crafted" linking suggest the necessity for additional means of linking for large scale projects or for converting existing text into hypertext documents. One of these methods is automated linking which uses keyword searches or techniques adapted from artificial intelligence. Firms are currently working to incorporate artificial intelligence into their authoring programs. "SmarText began this trend by performing automatic links on a document without any input from the author, except for the degree of linking that should be done." [Fersko-Weiss, 1991] Currently automated linking creates too many links that are not satisfactory, but the manufacturers are working on it. "While bad hits are more common with automatic linking, speed and thoroughness of connection can make compensation for the

inconvenience." [Alschuler, 1989] Structured linking is another method discussed below.

### 2. Converting Existing Text to Hypertext

It may be more cost effective to take much of the existing printed text and convert it to hypertext rather than rewrite it. There are many forms of text which lend themselves to ready conversion. Conversion consists of breaking existing text into nodes and linking the nodes in some structure.

Frisse [1988] identifies two distinct classes of links used when transforming a flat text file into a hypertext document. These are structural links and user-defined links [Frisse, 1988]. (Linking is discussed in-depth in Chapter III) The breaking up of text is easier for hierarchical documents. Existing document structure such as an index or table of contents can be used to parse the text into nodes and links which form a hypertext skeleton that corresponds to the hierarchical structure of the printed format. In these cases, nodes usually consist of the smallest labeled units in the text, for example sections, and the links are structural links [Frisse, 1988].

Even the most automated of conversion processes will likely require some customizing. This customizing takes the form of associated concepts the author sees as relevant to the current text. Customizing is also available to the user in the form of user-defined links which allow the creation of

nonsequential paths through the text and annotations to the text.

Frisse [1988] notes that it is rather simple to convert flat text files into a hypertext database. "You exploit document identifiers to parse the file and create the new data structure." [Frisse, 1988] Kellett [1989] echoes this sentiment saying "Training individuals to build hypertext documents using GUIDE is a simple process." What they are in effect saying is that the process of converting a flat text file into a hypertext format consisting of nodes and links is a mechanically simple process. The point they do not speak to is the type of functionality a user might expect from a system so easily derived.

After pointing out how simple it is to convert a document to hypertext format, Frisse acknowledges the difficulty with "Developing the software that lets you access appropriate portions of a hypertext document...." [Frisse, 1988] This reemphasizes the point that simply breaking the document into nodes and links does not make a useable hypertext system. There are likely to be numerous nodes required in the hypertext system which are not present in the printed document. The actual authoring process is much more complex than a simple parsing of the existing text according to the hierarchical structure of the original document.

The last point of converting existing text is the careful consideration of the type of document suitable for

this conversion process. Some forms of text do not lend themselves to the type of fragmentation required of hypertext systems.

> Fragmentation must be carried out carefully if the hypertext is to be a faithful representation of the original...structure must be inferred from a careful study of the text... there is always the danger that implicit, unrecognized structure will be lost when converting to a new representation. [Raymond and Tompa, 1988].

## E. COMPARING HYPERTEXT TO OTHER SYSTEMS

One of the missing links in the background material presented thus far is the practicality of hypertext systems as compared to other computer systems, more traditional forms of accessing text online, and to paper versions of text as well. We must not use new technology for technology's sake but rather to achieve some purpose or goal. For example when building a hypertext system to replace a printed version, we must be sure that as a minimum the electronic version can;

> ...duplicate the capabilities provided by the combination of an experienced reader and a well-designed paper text. Anything less degrades the system: leaving at best, an electronic page-turner; at worst, even less than the paper version. Furthermore, in order to justify abandoning the conventional method and medium, an electronic system should offer improvements, such as increased flexibility, reduction in storage, and convenient document development and maintenance. [Carlson, 1989]

### 1. Hypertext vs Other Computer-Based Systems

This section and the following section refer to a number of studies which will not be identified specifically but are referenced by Nielsen [1990a]. The reader is

encouraged to review this reference for further information concerning these studies.

One of the cited studies compared accessing text and comprehension via scrolling text files as in ordinary text processors and a hypertext version of the same information. The study showed that the users of the traditional system were able to answer the same 15 questions faster using the traditional system than the hypertext system (13.2 min vs 18.3 min).

Another study along similar lines had subjects read articles on a computer screen using either a hypertext format or the original linear format and were then asked to recall as many concepts as possible. There were two types of articles; general interest articles and technical articles. The readers of general interest articles performed worse using the hypertext system than the linear file system (17% vs 31% of concepts recalled) and about the same when reading the technical articles (22% vs 21%). Poor performance using hypertext might have been explained by the fact that the subjects received no training using hypertext and were consequently negatively biased towards it. This study might also indicate that the more highly structured technical articles were more suitable for a hypertext application.

One study compared a hypertext system to a more traditional menu selection system. The study found that the subjects performed better on the hypertext system and

subjectively preferred using the hypertext system [Nielsen, 1990a].

Many liken hypertext to a database concept, (See Chapter III) and thus some studies have been performed comparing hypertext and traditional command-based databases. They found that the total number of different nodes visited was roughly the same, but the hypertext system users revisited the nodes more often. The hypertext users had more rings and spikes in their navigation patterns than the command-based users. The researchers concluded that hypertext access capabilities allow users to move through the same data in different ways.

Hypertext systems and expert systems are often used in similar applications. One study ". .compared an internal IBM hypertext system to a commercial expert system shell for the representation of information needed to diagnose problems in a world-wide computer network " [Nielsen, 1990a] Users of the hypertext system correctly solved a larger percentage of the presented problems (81% vs 67%) but the expert system users were faster (5 min vs 4 min). Subjectively both authors and users of the systems preferred the hypertext version. Approximately 50% of the users would have chosen the hypertext system for the task versus about 25% for the expert system. The authors claimed it was easier to update the information in the hypertext system, than to maintain the knowledge base in

the expert system. The study was very limited and no other consequential results from comparing the systems were given.

### 2. Hypertext vs Paper: The Question of Linearity

One of the most important comparisons we can make is that of hypertext systems versus the traditional forms of printed text we are all so familiar with. The advantages of hypertext seem evident when the presentation characteristics of the printed material are inadequate. Texts that are poorly indexed, or which use too many layout conventions can cause frustration and confuse readers.

> Poor presentation characteristics particularly disadvantage books. Books cannot offer additional information beyond their "silent" text, and they provide only a limited number of entry points; nothing can be added to a printed text. Thus, users who have difficulty with texts have little recourse to resolve their difficulties. [Rubens, 1989]

Hypertext appears to resolve many of the issues addressed in this passage. In particular computer users want;

> ...to find information that will support their work and get on with it. They want quick entry points, identifiable information targets, and multiple support where necessary. They do not want to spend time reading long or complex blocks of text. [Rubens, 1989]

For this class of users, the measure of practicality for a hypertext system is in terms of how well the system enables them to do their job.

One of the studies compared a hypertext system with a paper implementation of the same information. The information consisted of a collection of historical articles which was

about 138 pages long in the printed version. Subjects were required to answer questions concerning the articles. The readers were faster using the paper version (42 sec vs 22 sec) when the answer was at the beginning of an article. They were marginally faster when the answer was in the body of the article (58 sec vs 51 sec). They took the same time when the answer required the combination of facts from two articles (107 sec). "These data seem to indicate that hypertext is of some help in situations where the user has to jump around in the information, and that hypertext slows the user down in situations where the information can be found by a glance on a page." [Nielsen, 1990a]

Another study had sixteen students use an encyclopedia in both hypertext and printed form. When asked to compare the systems, half said the hypertext system was faster. Three of the students said the hypertext version contained more information and one student said the hypertext system was more up to date. In reality the two versions contained the same information and the subjects were measurably slower with the hypertext system. This is a good illustration of the seductive powers of new technology, and the potential bias of subjective evaluations.

One study showed subjects were able to retrieve information to answer questions faster with a printed version of text when key words in the question were in the headings of the book. The hypertext system was faster when the questions

contained words from the running text of the book but not in its headings. Also, when additional measurements were taken concerning correctness of responses, the hypertext version was found to be better. One of the reasons found for the better hypertext performance was that the printed information did not highlight the more important information. Conversely, the hypertext system highlighted the user's search terms, pointing out a section which was particularly relevant.

There are very few studies which evaluate hypertext users' performance on a real world task. One such study was conducted in a British computer company where the task was to diagnose the fault when a customer called for service. The purpose of the task was to determine whether or not a repair technician needed to be dispatched and what spare parts were required at the site. The correct diagnosis had substantial financial implications due to the high expense of dispatching technicians to fix trivial problems or having the wrong parts. The study found that the hypertext system users correctly diagnosed customer calls more often than the paper system users (88% vs 68%). The hypertext system users improved their performance to 92% after six weeks of use.

Despite the fact that hypertext is billed as a nonlinear means of exploring an information space, there is much debate concerning the merit of linearity in a hypertext environment. It is also debateable whether "...hypertext is intended to overcome the deficiencies of linear print or of

sequential on-line text." [Alschuler, 1989] Alschuler further points out that:

> It may be that the "linearity" of print derided in the hypertext literature is less constricting for the reader than the blind loops and directional links of subjective hypertext. Unless mitigated by a number of clear choices, hypertext linking is not only "linear", but blind. [Alschuler, 1989]

The tremendous appeal of hypertext is most probably related to the way in which link traversals model the cognitive process of association. This gives the user the feeling of being in control, moving about the information according to his own desires. Jaynes poses an interesting question though;

> The question here, however, can be stated bluntly. Is this what readers... really want - or more importantly - really need? Is this the best way to communicate specific, instructional ideas between the writer (who's job, after all, is to help the reader understand on an intellectual rather than emotional level) and the reader (who has turned to the text specifically for intellectual enlightenment rather than entertainment or pleasure)?... the unstructured roaming encouraged by conventional hypertext systems is antithetical to the needs of both. [Jaynes, 1989]

In this passage the author is referring to instructional texts or user manuals associated with computers and associated software use.

The question of linearity is tied to the type of application the hypertext system is designed for. For some applications, any form of linearity in a hypertext system defeats the purpose of the system. For other applications, some form of guidance or linearity is almost essential.

The point of the above passage is well taken, and has direct application for the Department of Defense. (See Chapter VI.) Many of the applications that the DOD will need hypertext for more closely parallel the instructional texts Jaynes refers to rather than the "traditional hypertext" systems he criticizes.

## V. HYPERTEXT IN THE DEPARTMENT OF DEFENSE

### A. POTENTIAL APPLICATIONS

The broad range of specific applications addressed in the literature implies there are a number of applications suitable for the Department of Defense. When accessing the range of potential applications for DOD the first thing that comes to mind is to compare the volumes of instructions, policy guidance, maintenance and repair manuals, reference manuals, and other specific program guidelines with the three golden rules proposed by Schneiderman. Even a cursory review reveals that many of these DOD documents seem to contain the key attributes identified by the golden rules. For example, reviewing the Department of the Navy Automated Information Systems Security Guidelines (DON AIS Security Guidelines) reveals a very large document, organized into a strict hierarchy of chapters, paragraphs, and sub-paragraphs, which are all related to the concept or process of developing ADP security measures in accordance with Navy policy. Because it is used as a reference and guide, access to the document is primarily confined to a specific topic area, thus limiting the portion of the document required at any time. There are 26 chapters, two appendices, and multiple tables and figures. The manual has a 13-page table of contents and no index. It seems

very likely that the newly assigned ADP security officer who is unfamiliar with computers might find this document more that a little intimidating. In this case the possibility of online information retrieval might be very beneficial.

The DON AIS Security Guidelines is only one example from a myriad of documents which are similar in nature and would readily lend themselves to a hypertext implementation. There are numerous specific examples in the literature which implement hypertext systems for Department of Defense benefit. One of the early pioneering hypertext projects was ZOG, a computer assisted management system for the Navy's newest (at that time) nuclear powered aircraft carrier, the USS Carl Vinson. This was a much more ambitious program than a single document authored in a hypertext environment. ZOG supported four applications which included [Akscyn et al, 1988]:

- On-line policy manual (Ship's Organization and Regulation manual)
- Interactive task management system (for analyzing and tracking complex tasks)
- On-line maintenance manual with interface to video-disk (for weapons elevators)
- Interface to the AirPlan expert system (AirPlan developed at CMU by McDermott et al.)

These four applications give us some hints about general areas suitable for hypertext implementations. ZOG and KMS (the first commercial version of ZOG) have been found to be useful over a wide range of applications. Some of these include on-line manuals, on-line help for software, user interface to

other programs like expert systems, project management, electronic mail, financial modeling, accounting and operating system shells [Akscyn et al, 1988]. Many of these uses are applicable to the Department of Defense.

One use that particularly seems to stand out is the on-line maintenance manual. The complexity of equipment in the Navy has grown enormously and is continuing to do so. Documentation and maintenance manuals for these complex entities can be many thousands of pages and; "...locating relevant sections of the documentation to determine appropriate corrective action can be time-consuming." [Hayes and Pepper, 1989] The problem is made more difficult because this information is not static, but requires frequent update due to engineering changes, updates, corrections and additions. "Putting the documentation on-line can ameliorate these problems, both in terms of speeding the distribution of updates and in terms of locating relevant sections of the manual...." [Hayes and Pepper, 1989]

Another area which may be a prime target for hypertext applications is any complex program which relies on documents. Here complex can mean difficult to administer due to a large number of requirements, or it can mean difficult it understand due to technology or required specialized skills. An example of the former category might be the Command Managed Equal Opportunity (CMEO) program. This is an extensive program which has a very large number of requirements in order to ensure

compliance. It is not technically complex or generally difficult to understand but it has a large number of components which make it difficult to administer. It also very likely has a few areas which might benefit from expert advise. This suggests great potential benefit from expert system and hypertext integration. (See Chapter VI for more extensive discussion in this area.) The Navy's ADP security program might be a good example of a technically complex program. The background information in Chapter II outlines the difficulties encountered with this program. It will be further discussed in Chapter VI.

Information retrieval is another potentially broad application area within the DOD. "Large complex paper textbases exist in most professions...." [Carlson, 1989] Certainly the government, and in particular the DOD, is no exception here. Any administrative department in the DOD and has bookshelves full of volumes of related instructions. Most instructions contain multiple references. Sometimes these references are quite extensive. Each of the references is likely to reference other documents. A great deal of time is spent performing searches on these documents to retrieve information required in the daily performance of work. It has been estimated that 55% of the total paper carried on board ships is reference material [MacDonald, 1987]. No one can keep up with all of the rules and regulations pertaining to the way we must perform our jobs.

> The cost of retrieval remains very high. Time is valuable, and crew members using paper tend to ignore information which is not readily accessible. [Kellet, 1989]

Any type of system which simplified the access to this sort of routine information would be beneficial. Hypertext seems to be a natural fit for this application.

Here again, the union of expert systems or artificial intelligence with the hypertext system seems natural. This integration could be in the form of an expert librarian or "smart interface" which would be used "...to help the user select a path through the textbase that is tailored for a particular application or purpose." [Carlson, 1989]

> One example of an intelligent access mechanism for an aircraft manual is an online troubleshooting tree...a fully automated diagnostic system guides the technician to the most likely fault and then, on request, displays the manual text for the appropriate rectification procedure. In a sense, the automated troubleshooting tree becomes a filter for the textbase. [Carlson, 1989]

A technician's ability to quickly retrieve information from printed text depends on two things; the technicians understanding of the organization of the document, and the amount of experience they have had with the system [Carlson, 1989]. "Encapsulating user expertise and providing ease-of-access to electronic information are priority items for online information development." [Carlson, 1989] The biggest implementation problems for this type of system are probably the huge size of the required hypertext database and the current technology with information retrieval techniques.

Education and training is another area hinted at that may have solid applications in the hypertext environment. There is much emphasis in the literature placed on the idea of hypertext and its analogous relationship to human cognition; "...in particular, the organization of memory as a semantic network in which concepts are linked together by associations." [Shneiderman, 1989] The contention is that if this is true;

> Learning theory would predict that hypertext should improve meaningful learning because it focuses attention on the relationships between ideas rather than isolated facts. The associations provided by links in a hypertext database should facilitate remembering, concept formation, and understanding. [Shneiderman, 1989]

There is a good deal of literature which discusses the use of hypertext in CAI applications. The use of hypertext programs for training environments in the Department of Defense seems a natural application. The relationship between hypertext and CAI will be discussed further in Chapter V.

One of the first practical implementations of hypertext was the ZOG system referred to above. There are a number of other specific examples which use hypertext technology to address some need in the DOD. The following paragraphs describe a couple of these systems. They are important because they provide insight into the capabilities of the technology and insight into the design characteristics required for useful hypertext systems.

Hayes and Pepper [1989] describe an Integrated Maintenance Advisor (IMAD) system which is designed to support fault diagnosis in the HAWK missile system used by the U.S. Army and other NATO countries. Their work presents a "...framework for supporting fault diagnosis in complex systems." [Hayes and Pepper, 1989]. "The frame work is based on a marriage of diagnostic expert systems with a hypertext representation of written documentation, supplemented by conceptually-based text processing techniques." [Hayes and Pepper, 1989]

This system is an excellent example of the online maintenance manual mentioned earlier. It provides insight into the enormous potential benefit derived from the merging of these technologies. Hayes and Pepper cite advantages IMAD provides over existing approaches [Hayes and Pepper, 1989]:

- Use of a hypertext representation for documentation without the other two technologies will result in a system where the documentation is easier to browse than hard-copy documentation, but where ease of location of relevant information will be little improved. Moreover, such a system lacks the expert assistance possible through diagnostic expert systems.

- Use of diagnostic expert systems without a full representation of the documentation cannot serve the needs of complex systems because of practical limitations to current diagnostic technology. Moreover, such a system does not allow a technician to browse for greater understanding or insight into the system being maintained.

- A system which merged a documentation hypertext with diagnostic expert systems without use of conceptually-based text processing techniques would face major problems in the construction and maintenance of the necessary documentation indexing and of the cross-linking between the documentation and diagnostic subsystems. Both kinds of links are essential to proper functioning of the arrangement and to be practical, they must be constructed

automatically. Only by using conceptually-based techniques, can such automatic construction be made accurate enough.

Another application which integrates expert system technology and hypertext is described by [Lafferty et al, 1990]. This system, known as the Explosive Hazards Classification Expert System, has a hypertext database based on the Department of Defense Explosives Hazards Classification Procedures (also referred to as TB 700-2). Although based primarily on the TB 700-2, explosives hazard classification entails expert judgment not captured in the original document.

> In its present form, the TB 700-2 contains fundamental information about the testing and classification of explosives. It provides a good introduction to the process of explosive hazards classification and it is a good reference tool. However, TB 700-2 does not capture the nuances of hazard classification. ... In addition, the experts who routinely classify substances know much more than is contained in the manual. Because substance testing is expensive, the experts routinely classify new materials by comparing them with similar items whose categories have already been determined. This is termed "classification by analogy." TB 700-2 does not address classification by analogy. [Lafferty et al, 1990]

In order to address issues raised by this problem, the system developers "...envisioned a system that would provide hypertext access to both TB 700-2 and expert knowledge beyond what is contained in the manual and which would apply expert knowledge to properly classify hazardous materials." [Lafferty et al, 1990] The developers recognized some deficiencies in the manual and identified a group of experts who agreed to help develop the expert system rule base and provide supplemental information for the hypertext module. The

developers maintain that, for a number of reasons, "...this approach complicates development and puts more responsibility on the user, but it produces a much more powerful and useful system." [Lafferty et al, 1990]

This type of system is a good example of a technically complex system that is based principally on documents. The important points of this example are; the idea of supplementing existing documents with expert knowledge, the arrangement of the hypertext database according to different views (This will be discussed further in Chapter VI.) and the integration of expert system and hypertext technology.

## B. POTENTIAL BENEFITS AND ISSUES

One important clue that might indicate the potential strength of hypertext is the notion of Document-Based Decision Support Systems referred to by Keen [1987]. According to Keen;

> For anyone who knows nothing about computers, information mainly means documents; many aspects of work center around purchase orders, manuals, contracts, discursive reports, etc., etc. Data base management systems focus on only a small subset of organizational information activities and for many managers, the ability to scan, cross-reference and locate documents may provide far more payoff and involve much simpler systems than those that manipulate numerical data bases. [Keen, 1987]

Keen further talks about traditional DBMS being more restrictive than the ability to "...begin from a broader perspective of classifying and storing whole documents which are the unit of information and mental frames, zero in on interesting items, read and think and then scan deeper or

across the information base." [Keen, 1987] Keen concludes his discussion of document-based DSS saying;

> This opens up a new form of traditional DSS support: it balances computer power and managerial judgment. It may be limited by the generally hierarchical structuring of the information...but all in all it seems clear that the entire computer field is shifting to a document-centered world and DSS builders should exploit the opportunities this opens up. [Keen, 1987]

It seems most remarkable that Keen could write this article in 1987 with no mention of the term hypertext; for he seems to be describing the very essence of hypertext and one of its most fundamental uses. Hypertext is a natural way of achieving what Keen was describing. The fact that he essentially describes hypertext with no mention of the term serves to further emphasize the relative "newness" of the technology despite its rich history.

The above passages might prove to be quite visionary. Anyone working in the Department of Defense can relate to the importance of documents in accomplishing work. Many of these people have limited computer experience so the concept of information as documents is very relevant. Virtually all work done in DOD is document based. Documents are referenced routinely for almost every aspect of work. For any procedure there is some instruction or reference manual existing which details that procedure. Ships and aircraft are operated and maintained in accordance with documents. Personnel administration uses documents, policy is implemented through documents, tactics are executed according to documents and

personnel are trained using documents. DOD uses documents! Often users require access to multiple documents at the same time to perform work. Finding the relevant information can be time consuming. Any tool, such as hypertext, which allows quicker access and information retrieval from these multiple documents has implications for greater efficiency and improved job performance.

Most people would likely agree that society is experiencing a technological and information explosion. Nowhere is this more apparent than in the Department of Defense. For example the computer code in a Vietnam War-era F-4 was virtually nonexistent whereas there are some 236,000 lines of code in a more current F-16D. The code for the B1 is estimated at 1.2 million lines [Kitfield, 1989]. Commensurate with this explosion in software complexity is the volume of related documentation. "For example the documentation for an F-18 fighter aircraft is 300,000 pages big and requires 68 cubic feet of storage space...." [Nielsen, 1990a]

This huge volume of information suggests another potentially enormous benefit to hypertext. The aforementioned 68 cubic feet of storage can be contrasted with the just .04 cubic feet required to store the same information on a CD-ROM hypertext [Nielsen, 1990a]. The savings in storage are easily imagined. "If just the static reference material carried on a ship were stored electronically, a significant amount of paper either could be removed from ships or, at the very least,

could be removed from logrooms to fireproof storage spaces as a backup in case of power failures." [Kellet, 1989] In fact Kellet [1989] argues that hypertext may be a significant step towards the idea of a paperless ship in the Navy.

As mentioned previously, documentation is not static. Putting documentation on-line not only saves in terms of speed of updates, it can also realize huge potential cost savings associated with these updates. Concerning the F-18 documentation;

> Imagine the mailing costs involved in shipping updates to Air Force bases around the world by classified courier service. And imagine the scenes...when every single updated page has to be inserted in the right location in the right binder. Instead one can just press a new CD-ROM and tell people to destroy the old one. [Nielsen, 1990a]

The potential savings must be projected over the lifetime of the documentation. One must remember that updates are not an uncommon occurrence. "Boeing...issues a full set of revised documentation every 90 days." [Hayes and Pepper, 1989]

Another strength of hypertext is in its general appeal as a user interface. The concept of feeling in control and moving about the information space according to a train of thought instead of being a slave to the computer program may be more appealing to many users. Shneiderman [1989] hypothesizes that this greater sense of control may produce increased desire to explore further. "In the same way that computer games can be very absorbing because of the high level of interactivity, hypertext databases may be very engaging too." [Shneiderman,

1989] Kellet notes "Hypertext is a desirable retrieval system because it is simple, friendly, and has a familiar look and feel." [Kellett, 1989]

The question now is whether in fact hypertext is any good, and if so, does it have a place in the Department of Defense? The answer to the first part,

> ...is simply, "It depends",since it seems that some studies indicate advantages to hypertext while others indicate disadvantages. It depends on the hardware and system software used, it depends on the design of the hypertext system, and it depends very much on the user's task and individual characteristics. [Nielsen, 1990a]

In order for the system to gain acceptability and be used, it must demonstrate clear advantages over the more traditional paper versions people are so used to working with. To simply convert printed material to electronic versions of the same material is not good enough! Extra value must be added to the system. This will be further addressed in Chapter VI. When possible, the advantages should be measurable both in performing work and in cost savings.

The number of potential application areas outlined in section A seems to indicate that hypertext may have a very important place in the Department of Defense. The key to successful implementation of hypertext systems lies in the design of these systems. Personnel in DOD are primarily task driven in the use of computers. The computer is a tool for supporting the primary mission and not an entertainment device for intellectual exploration. As such, users require a tool

which most effectively supports the task at hand. Users are less likely to require or even have the patience for exploration, but need instead to be lead or guided to the solution they are seeking for a specific problem. If the hypertext systems built do not keep this in mind they are likely to be a "...condemnation, not a solution." [Jaynes, 1989]

Hypertext has certain current technological limitations. Not all applications are suitable for hypertext solutions. Before blindly applying hypertext technology to all problems, a careful assessment of its capabilities or benefits, potential problems or liabilities, and the desired goals achievable through application of this new tool must be done. There are a number of important criteria which must be applied to judge whether an application is suitable for a hypertext solution. The following questions should be addressed before deciding to use hypertext to solve a problem:

- Does the document or set of documents envisioned for hypertext application satisfy the "three golden rules" of hypertext?

- Does the application lend itself to a computer solution? In other words "...do not use hypertext if the application requires the user to be away from the computer." [Nielsen, 1990a]

- Is there a problem with information retrieval using the current system? Will a hypertext implementation of the document help the problem or magnify the problem?

- Does the user task require guidance or advice based on documents? (For example, maintenance publications)

- Will a hypertext implementation offer more utility than the current system? (There must be incentive for users to switch.)
- Are there potentially large cost savings in terms of storage, update, and document preparation?
- Will the end user group accept and use the technology? (Nielsen found that the single greatest factor affecting the useability of hypertext was the "individual differences among users." [Nielsen, 1989] The younger the user population the more likely they were to accept and use the technology.

What seems important to note is the potential enormous benefit derived from the addition of artificial intelligence and/or expert systems to almost any type of hypertext system. The combination of hypertext and expert systems seems to be a highly synergistic means of handling a large variety of problems within the Department of Defense. Development of this type of system should be highly encouraged.

# VI. HYPERTEXT AND THE DON AIS SECURITY GUIDELINES

Chapters III and IV provide the background information necessary to evaluate potential applications for hypertext systems. Chapter II suggested that a hypertext system might be built using the DON AIS Security Guidelines as the underlying document. It was suggested that such a system could improve information access and retrieval from the document, provide supplemental information in difficult to understand areas, provide immediate access to a glossary for unfamiliar ADP terms and acronyms, provide access to additional reference material related to the specific task, provide a tutorial lesson in the difficult to understand areas, and even offer Expert system advise for some procedures.

## A. THE IDEA OF VIEWS

To merely convert the DON AIS Security Guidelines into a hypertext document seems to have little practical value and might be a bad idea. Many technical manuals typically have flaws. For example,

> Some sections are incomplete, ambiguous, or presume too much from the reader, and the documents's organization sometimes hinders understanding. Simply scanning the document into hypertext is not a good way to convey its meaning... [Lafferty et al, 1990]

As illustrated in the case study, the DON AIS Security Guidelines has a number of these flaws and may not be a good

candidate for hypertext conversion in and of itself. In order to have practical benefit the system must have something of value added to it. The added value should be substantial in order to overcome some of the inherent deficiencies of hypertext as compared to paper. The capabilities of the system proposed above suggest a great deal of added value to the system. It would integrate a number of technologies which include computer aided instruction, expert systems, artificial intelligence and hypertext. The capabilities of the suggested system also give clues concerning design issues. For example the use of implicit links is suggested by the reference to a glossary requirement.

Another form of added value to the program is the possible reorganization of the information to support multiple functions. The great "...advantage of hypertext is that it facilitates alternative organizations of information." [Lafferty et al, 1990] This suggests the concept of views: "...hypertext offers the possibility of creating many views on the same set of underlying fragments." [Bruza and van der Weide, 1990]

For example the DON AIS Security Guidelines could be organized according to different views based on what the user wanted to do with the system. This is similar to one of the operational advantages of hypertext noted by Conklin; "the idea of customized documents in which text segments can be threaded together in many ways, allowing the same document to

serve multiple functions." [Conklin, 1987] There could be an information retrieval view which allowed a user to access the document to locate information to support a specific task. There could be a diagnostic view, which allowed the user access to a set of expert system advisors for security related issues. There could be a program implementation view which guides users through the steps required to implement an AIS security program. This type of view might even be tailorable such that the user could input the type of command including the related computer assets and the system could present the information relevant for that type of command. Lastly there could be an instructional view, which could be used to provide tutorials and intelligent instruction for DON security related issues.

Constructing views would limit the amount of information the user would have to wade through and confine the information to only that relevant for the user's task. It might even be possible to filter the views according to the users relative expertise, showing more extensive information and having more guidance navigation mechanisms for novices than for experts. This is important because it is likely that different users will want to use the system in different ways. This notion of views will be elaborated further later in this chapter.

## B. INTEGRATION OF OTHER TECHNOLOGIES AND HYPERTEXT

There is plenty of evidence to suggest that the best utility from hypertext systems will come from the integration of hypertext with other technologies. The area of computer security is one of the application areas within the Navy which might fall into this category.

The following sections discuss the integration of these technologies and describe the design features of a possible hypertext system which could be built to address some of the needs outlined in Chapter II.

### 1. Expert System Integration

The marriage of hypertext systems with expert systems seems very natural. As noted in Chapter V, there are a number of existing systems which already incorporate this idea. There are several ways for expert systems and hypertext systems to be integrated. Carlson [1989] delineates at least three possible ways to conjoin expert systems and hypertext.

The first way is to have the knowledge base and hypergraph as separate components of a system. "This configuration uses the knowledge base as an interactive interface to filter the information-rich chunks of information in the hypergraph." [Carlson, 1989] After a consultation with the expert system, the appropriate segment of the document is accessed.

A second method is to merge the knowledge base with the hypergraph. This method can be used to overcome one of the principle criticisms of expert systems. They are often criticized as being;

> ...too brittle; that the user is led, lock-step , through a series of information-extracting questions without adequate opportunity to interject intuition or to adapt the system to the particular situation. Eventually, the user begins to feel like a slave. In fact, the educational value of an expert system is limited if the user cannot understand the rationale behind the various steps in the consultation session. [Carlson, 1989]

The proposed solution is to design the knowledge base as a hypertext [Carlson, 1989]. The result is a seamless highly modular environment. KnowledgePro™ is an example of this type of integration. Using this system the expert;

> ...builds text blocks in a manner similar to how she would sit down and tell someone what she knows. The expert also links these chunks of knowledge to others. Eventually, the expert adds rules and an inference engine to boost the intelligence of the system. [Carlson, 1989]

The third possible approach is to embed or distribute expert systems within the hypergraph. "The hypergraph retains its integrity, but the user is able to call up search aids at given choice points in the text base or the search aids may act as demons which awaken given a particular set of circumstances." [Carlson, 1989] Although referring to intelligent interfaces for text retrieval, the idea can be extended to include expert system advisors embedded within the text base [Carlson, 1989].

Although each of these options might offer a viable solution, the last solution seems to offer the best fit for the system this chapter proposes. The first solution is possible but may suffer from the weaknesses pointed out for expert systems. The second solution does not take advantage of the information already available, requiring extensive reauthoring. There is another potential weakness as well;

> Expert diagnostic systems are highly effective with relatively small, focused applications (i.e., where the knowledge base contains up to several thousand rules or objects). However, because the construction and maintenance of knowledge bases is still a labor intensive process, this approach can be inappropriate when applied to very large, complex systems...Moreover, even if extremely large diagnostic systems could be constructed, the uniform interpretive mechanisms they employ would represent overkill in many simple situations... [Hayes and Pepper, 1989]

The third solution, where a number of relatively small expert advisors are distributed throughout the hypertext database seems most congruent with this application. There are many sections of the DON AIS Security Guidelines which do not require expert advise to interpret. Certainly though, there are a number of sections which would benefit from immediately available expert system advise. For example, the entire chapter on Risk Analysis would benefit from this advise.

Risk analysis is a complex task that is very intimidating to the inexperienced security officer. There are multiple methodologies, none of which appear to be straight forward. The Trusted AIS method of risk analysis is mentioned, but not described thoroughly because it represents a new

methodology under development. Yet the reader is lead to believe that this may be the most practical for many types of commands. Method selection is based on;

> ...the complexity of the AIS environment. The complexity of the AIS environment is governed by the level of data processed, security mode of operation, AIS system configurations and locations (stand-alone, networked), and the criticality of the mission. The physical makeup of hardware...involved is not a determining factor in method selection. [DON AIS Security Guidelines]

Not only are the criteria not obvious to an inexperienced security officer, the guidance given to make a method selection is not clear.

> Method I is the standard method for use in most AIS environments. Method II is for use in less complex AIS environments. The Trusted AIS Method is a new methodology currently under development....With this method, any AIS can use the same evaluation criteria. [DON AIS Security Guidelines]

Pity the poor inexperienced security officer who must base a decision on this material! None of the components which determine the complexity of the AIS environment are straight forward. They each require elaboration for clarification. Furthermore none of them are even referenced in this particular section. The reader must scan the document on his/her own to find the clarifying material. Even if the security officer understands the complexity of the environment, the selection criteria are very vague. Should the officer decide that the Trusted AIS Method sounds the best, there is limited guidance.

> This method...involves a three-phased approach which requires the determination of Required Operation Trust

> Evaluation Level (ROTEL), evaluation of the computer system based on the Department of Defence Trusted Computer Security Evaluation Criteria (DOD TCSEC), and identification of operational controls to satisfy mandatory minimum requirements as specified by reference (a). [DON AIS Security Guidelines]

Carefully analyzing the information presented thus far points out several deficiencies a security officer faces using the printed material which might be corrected with the implementation of an expert/hypertext system. One deficiency is that the cited material references several additional documents which the reader must either already understand or look up before proceeding. The material also mentions terms which might require further clarification. A hypertext implementation would allow the user to link to the appropriate portion of the cited reference to either refresh or learn the applicable information. It might also allow the user to link to the components of the AIS environment which are not elaborated in the risk analysis chapter to gain the necessary explanation.

Another weakness noted earlier is that the decision criteria for risk analysis method selection are very subjective and unclear. An expert system might be of great benefit with method selection and might also limit the amount of additional research necessary to perform this procedure. Further analysis of the Trusted AIS Method shows that calculating the ROTEL is a complex five step procedure which requires the use of four different tables and matrices to

calculate. Examining step two shows us the following explanation;

> Determine the Communication Path Capability. The communication path refers to the method in which the user transmits and receives information to and from the main AIS (CPU). Interactive terminals that are connected to a host, either directly, through a local area network, or through a long-haul packet network, are more vulnerable to penetrations than those connected only through a store-and-forward network. This is because of the increased bandwidth and closer host-terminal interaction they permit. [DON AIS Security Guidelines]

This passage is likely to be completely unreadable to the novice computer security officer. It is rife with terminology which will not truly be understood by even a relatively sophisticated home computer user. This might be another area where a small embedded expert system would be useful. It would weed out the irrelevant information and speed up the implementation process. With no expert advise, this passage might require extensive research simply to understand.

Embedding the expert systems into the hypertext database seems to be the most dynamic and flexible solution as well as providing the best user interface. Although the evidence is sparse, the studies mentioned in Chapter IV seem to indicate users prefer a hypertext interface to an expert system interface. Hypertext also seems to offer greater flexibility with regards to updating the information space. Subjectively, it seemed easier to update the information in the hypertext system, than to maintain the knowledge base in the expert system. The ability to link anything in a hypertext

system should allow for a more seamless interface between the reference material, the expert system advisors, and the instructional material.

### 2. CAI System Integration

One of the common threads in the research for this thesis was that each technology had applications espoused which were very useful for learning. The potential of hypertext for learning has already been touched on in Chapter III. The relationship between CAI and learning is obvious. Generally "the major objective of an expert system is to render advice, whereas the purpose of CAI is to teach." [Turban, 1990] However, "many expert systems can be used as tutors in addition to being advisors." [Turban, 1990] Expert systems have also been used for training via their explanation facilities. There is ample literature concerning the conversion of an expert system to an intelligent CAI system. Bevan and Wetherall [1987] discuss some of the issues involved with this.

The potential use of all these areas for learning suggests that their successful integration might yield a very capable instructional or training system. This instructional capability may in fact be only a side benefit from the specific application the system was designed for. Integration of these technologies based on a hypertext system which uses

the DON AIS Security Guidelines as a foundation could address some of the training deficiencies noted in Chapter II.

There has been a lot of excitement concerning hypertext and computer aided instruction. The ability of hypertext to link to almost anything is beneficial for instruction in that it allows a variety of media to be applied as appropriate to the learning situation. As mentioned in Chapter III, the organization of memory as a semantic network in which concepts are linked together by associations suggests hypertext should be a good learning tool. This however, does not mean that it is a good idea to allow completely free exploration of a network of nodes and links in a learning situation. Hypertext must have features which support and facilitate learning [Jones, 1990].

> Just as there are many models of hypertext, there are many theories of instruction, and a number of categorizations of learning. Therefore, it seems reasonable to expect that different forms of hypermedia systems will be effective for different purposes. [Jones, 1990]

The intent here is to illustrate that hypertext and CAI can be effectively integrated together and that there are multiple considerations when doing so. This thesis will not propose the best way to integrate CAI and hypertext, but rather assume that it can be integrated and that this is a potential benefit to the proposed system.

Using computer based instruction of security procedures within the Department of Defense is not a new idea. For example, the Air Force has developed a <u>Computer Based</u>

Instruction for Computer Systems Security Officers for the Air Force Cryptologic Support Center (AFCSC) at Kelly AFB, San Antonio, Texas. This system solution was an attempt to provide cost effective security training to a large number of dispersed Air Force personnel. Much of the work done for this system might be directly transferable to a hypertext based system for the Navy. It might also be possible to use the training material used in a number of security courses offered within the Navy to generate an instructional component of a hypertext based system with nominal effort.

## C. SYSTEM DESIGN

Finally, it is time to look at the overall system and some features required to make the system practical and useful. The design considerations reviewed in Chapter IV can serve as a framework to help establish the proposed system features.

### 1. System Users

The needs of the users are what must ultimately drive the system design. Here we must assume that the users will be task driven. Most users will be required to implement an ADP security program for their respective command. The proposed system must support the many tasks these users must do in order to achieve that goal. Examples of these tasks include, but are not limited to, ensuring individual accountability, ensuring physical control of computer resources, certifying system accreditation, conducting risk management, and

conducting security awareness and training. SECNAVINST 5239.2 and the DON AIS Security Guidelines delineate and amplify these tasks.

The task driven user needs quick access to only relevant information. "The user's fundamental need is for explicit procedural and reference information accessed in predictable ways." [Herrstrom and Massey, 1989] The users need to be guided in some sort of way. They do not need or have the time for exploration. This implies that certain navigation and information retrieval tools are required. The availability of these tools is dependent on the database structure.

The section addressing the problem of risk analysis noted earlier has some implications for the user's needs in this area. The passages clearly indicated the need for an inexperienced user to be guided through the information. A tool that simply brought the user to the applicable section of the reference material would be sorely deficient if the information presented was unintelligible. In the risk analysis example, the user would need to be guided through the environmental analysis, the choice of methods, and the steps required to implement the selected method. In general users will need to be guided through the different views available to them in order to perform the tasks the views were designed to support. Related information is examined further in the section which discusses navigation.

### 2. Database Structure

"As a generalization, it seems that engineering-oriented hypertext users prefer hierarchical organizations, whereas arts-or humanities-oriented users prefer cross-referencing organizations." [Conklin, 1987] Users in the Department of Defense generally fall into the engineering category and by extension are more likely to be more comfortable with a hierarchical structure. Using a hierarchy allows for exploitation of the underlying documents' already existing hierarchical structures and will help give users the more familiar look and feel they get from working with printed material. As noted in Chapter IV, this sense of familiarity may be important to gain user acceptance of the system as the transition is made from a paper medium to an electronic medium.

There may need to be several hierarchical structures though. The potential need for different views of the underlying document may require the organization of the database according to several hierarchical categorization schemes which each support the different views discussed earlier. For example the structure of the view required to support information retrieval may have material different from the view required to support instruction.

Some of the problems endemic to hierarchical structures noted in Chapter IV might be mitigated by multiple hierarchies. Despite multiple arrangements of the information,

the fact remains that the important criteria which define these structures must be anticipated.

A second potentially more powerful solution to the problem of supporting multiple arrangements of the data is the notion of virtual structures introduced by Halasz [1988]. This notion of virtual structures is an adaptation of the concept of views in a relational database system. In order to support this idea, hypertext systems would require "...a substantial search query mechanism over the hypermedia network." [Halasz, 1988] This type of mechanism would allow a user to form large composites of the information contained in the database from a query which defines the components of the virtual structure.

Currently there are a number of technological problems with implementing this type of solution. The practical problems in constructing views include the notions of redundancy and irrelevance. These are the criteria by which we can judge views [Bruza and van der Weide, 1990].

> If there is too much redundant information in a view then this is annoying for the user. Similarly, if a view is based on a certain theme and there are too many irrelevant aspects with respect to this theme, then this is tiresome for the user. [Bruza and van der Weide, 1990]

While this solution is not practical in terms of today's systems, it may be a possibility in the near future.

The specific implementation of an ADP security program is different in most Navy commands because there are a number of unique circumstances present in each command which have some bearing on program implementation. For example, on one

extreme are small commands which have a limited number of stand-alone micro computers, processing sensitive unclassified information in a limited access security mode. On the other extreme lies those commands which possess huge and complex systems which incorporate the use of mainframes, mini-computers, and micro computers with all of the associated peripherals in a networked environment, processing all security levels of information and operating in a partitioned or multilevel security mode.

The DON AIS Security Guidelines is an extensive and comprehensive document designed to address both circumstances outlined above as well as the entire spectrum of possibilities which lie in between. As such it is generally unsuitable for very many specific applications. The idea of predefined hierarchical organizations or virtual structures of the database may be one way to tailor the document to individual command needs. It may be possible to put all the information necessary to implement an ADP security program in any naval command in a single hypertext database. The trick then would be to filter the database to collect all the relevant information into composite structures which represent the information necessary for an individual command. These composites would then have to be filtered, possibly more than once, to eliminate redundant and irrelevant information. This might prove to be more difficult than practical.

Although each command has very specific needs, there are certain basic elements common to every program. Similarly each command has to go through some version of the same elemental steps in order to implement a program. This suggests that each command will have a need for the same basic views noted earlier. This further suggests that a number of predefined views or structures, such as the ones noted in section A, might be a good first step towards customizing the database to suit user needs. However, there may need to be an additional filter on these views based on some command specific information such as security level of information processed, system user's clearance level and need to know, system configuration and location, and user capabilities. This filter could be imposed on one of the original predefined structures to further refine the information available to the user and hopefully customize the database to better fit the user's needs. The use of artificial intelligence to assist in this filtering process would almost certainly be required. A computer novice should not be trusted or required to structure a view adequate for their own use. An expert advisor or AI interface to the query/filter mechanism would be necessary.

An example of this could be to look at one of the extremes outlined above. In the less complex situation there is a large portion of the document that is completely irrelevant. For example, an ADP security officer in this more simple situation would never even need to know about the

method I risk analysis methodology. The less sophisticated users should never have to go through complex calculations in order to arrive at the conclusion that they need to implement security measures which comply with the C2 level of security. Also, there are a number of manual solutions which satisfy the minimum mandatory requirements for the C2 level of Trusted Computer Security Evaluation Criteria. The less complex situation would likely employ a greater percentage of these manual solutions and therefore these solutions should be especially highlighted for these users. This is the type of information which should be somehow collected into a composite structure and presented to this class of user.

When implementing an ADP security program, the security officer would initially call up the implementation view which contained only the nodes required for program implementation. He/she could then structure a query or be guided by an AI interface which would allow him/her to put in some command specific information. This would further refine the view to eliminate the nodes which contained the information concerning method I risk analysis and other similar data, and maybe highlight information which might be particularly germane. In this way the user is left with information applicable to his/her own situation, greatly simplifying their ability to implement a program which complies with the original instructions. While the information might not be refined to a command specific level, the use of

embedded expert systems in conjunction with this reduced database might greatly simplify the ADP security program implementation process.

### 3. Linking

In order to support the hierarchical structure of the database, organizational links derived from the hierarchical structure of the original documents should be the primary explicit links used in the system. The table of contents can be used to parse the text into nodes and links which form a hypertext skeleton that corresponds to the hierarchical structure represented in the printed format. These could be automatically derived after the document has been converted into electronic form.

Additional forms of automated linking should be used as well. The problems with hand-crafted hypertext were discussed in Chapter IV. Linking for this system as well as virtually all Department of Defense applications should be done as automatically as possible. Automated linking based on context or a conceptual basis is important for the system. An example of this is the Text Categorization Shell developed by the Carnegie Group. This tool allows;

> ...the appropriate references to be found and index entries made on a conceptual basis, rather than based on the occurrence of specific key words and phrases. This insensitivity to the way that potential references are phrased makes it possible to generate almost all the appropriate references automatically (high recall) while minimizing the number of irrelevant references and index entries (high precision). [Hayes and Pepper, 1989]

Even the most automated of linking procedures will need some form of customizing. This will require an expert or a team of experts to link additional relevant annotational nodes and reference material to the existing hypertext database. This is one of the greatest strengths of hypertext. As noted in Chapter IV, a poorly presented book offers no material beyond what is in print and nothing can be added to it. A hypertext system can have an unlimited amount of additional reference material beyond what is in the original document. This will be further discussed in the next section.

The use of implicit links in the system is also a good idea. One of the problems with the DON AIS Security Guidelines noted in Chapter II is the large amount of confusing terminology and acronyms. Implicit links are well suited for this type of problem. Implicit links are also necessary for information retrieval, and information retrieval based on context is generally better than information retrieval based on keyword search. It has been noted that navigation tools alone are not enough to guide a system user to the relevant information they require. Implicit links are required to support the idea of system query or view filtering noted above. These structures must be computed via some mechanism which activates an implicit linking scheme to compute the new view or structure.

### 4. Database Components

One thing that may not be clear so far is the actual documents or sources of information which should be used for the hypertext database. It was stated earlier that simply converting the DON AIS Security Guidelines to hypertext format is not sufficient. The DON AIS Security Guidelines is just one obvious component of the system. The SECNAVINST 5239.2 defines the requirements for Department of the Navy ADP security programs and also is an obvious candidate. Enclosure (3) of the SECNAVINST 5239.2 lists 42 documents related to implementation of an ADP security program. Portions or all of many of these documents might need to be incorporated into the database. This will require expert judgement which is beyond the scope of this thesis.

Another source of information for the database is an already automated ADP security program which is comprised primarily of text files. An example of this type of system was noted in Chapter II. This system has many sample forms which could be placed into nodes in the hypertext database and linked where appropriate.

Lafferty et al [1990] give additional insight into source information for the hypertext database. When building an expert system for explosive hazard classification, they provided system access to a hypertext version of the existing classification manual. This manual was supplemented with expert knowledge beyond what was contained in the original

manual. This seems like a good idea for almost any manual. Supplementation of the original manual with expert knowledge might help mitigate some of the weaknesses common to technical manuals noted earlier. This is the type of information well suited for annotational links used to customize the database.

As alluded to earlier, the DON AIS Security Guidelines has numerous sections which are difficult to understand or assume too much knowledge on the part of the reader. An example of this used earlier is step two of the ROTEL procedure. This section would be an excellent candidate for supplemental information which translates the material into verbiage more intelligible for a novice.

Lafferty et al gained the additional information they required by identifying experts willing to help supplement the manual and conducting interviews with these experts to resolve unclear areas within the manual. Similarly, security experts within the Navy could be identified to help translate and supplement difficult sections of the guidelines. One good source might be the personnel who teach ADP security courses at the Navy Regional Data Automation Centers. Anyone who teaches has a good idea where the most common problem areas are. These instructors could be used to identify and augment deficient sections of the manual.

### 5. Nodes

One of the more complex design issues is the construction of the nodes. This thesis will not presume to dictate a particular style for the proposed system but instead offer some general insight and recommendations.

Of the numerous systems studied during research, no single data model stands out as the best one. In general, the data model presented by KMS seems to be as good as any. The idea of each node as a screen sized work space which can contain anything from text to procedures seems congruent with the way personnel in the DOD are used to seeing information in printed text. Because we want to embed expert systems into the hypergraph, it is important that the nodes can contain anything. One of the problems of hypertext is that to date "...virtually all systems have been insular, monolithic packages that demand the user disown his or her present computing environment to use the functions of hypertext...." [Meyrowitz, 1989] Hypertext systems within the DOD must allow for a more extensible environment which incorporates multiple technologies.

### 6. Navigation Tools

One of the most important features determining the useability of the system is the set of navigation tools available. As noted earlier this is a task driven system which requires that the user gain access to relevant information as

quickly as possible with minimum need for unguided exploration. The use of paths or guided tours through the different views should be incorporated as much as possible to allow the user to access relevant material in predictable ways. This might be especially useful in the program implementation views and instruction views of the hypertext database described earlier. A good possible solution is the path mechanisms described by Zellweger [1989].

The notion of some form of guidance navigation tool may be especially important for this system. One of the problems noted in the case study was that there is limited computer experience in many Naval commands both in terms of using computer technology and in terms of understanding computer security. This has design implications for the system. The user interface will have to be friendly enough for the novice to use comfortably. The same is true for the navigation and information retrieval tools. Imagine how the inexperienced ADP security officer might feel if he/she were given a complex and intimidating computer program and told that it would solve their ADP security problems! Imagine the feelings of the computer novice, looking at a hypertext database consisting of thousands of nodes and having the system tell him/her to enter the database and explore! Most surely the system must provide simple to use navigation and information retrieval tools which guide the readers to the information they require.

In addition to the guidance tool, some form of graphical browser is also probably a good idea. Despite the fact that guidance should be provided, users will doubtless want to branch off the path or tour from to time to time. Also certain views will likely be more conducive to exploration than others. Some form of tool that allows orientation within the database is therefore essential. Additional navigation tools should include a backtrack feature as a safety feature for disoriented users, and historical tools to assist in relocating previously found information.

The navigation/information retrieval tools available in the system could be used according to the level of detail the reader was working in or the view in which the reader was working. Paths could be provided to allow access to the several predefined hierarchical structures cited above. Additionally, some form of user defined query could be used to construct a virtual view of the hypergraph according to the individual's need. An information retrieval tool, possibly compatible with the query tool used for view construction, could be used to locate specific relevant information within the newly constructed virtual structure. This specific information might be in the form of composite nodes which were filtered by the query. Once located in a composite node, additional navigation tools could be used to explore the local area according to user desires. These navigation tools would

address the disorientation problems and the "context in the small" problems.

## D. FINAL REMARKS

Solving the problems associated with ADP security is critical. The fundamental problem with the notion of implementing an ADP security program in every command in the Navy is that each command is given a single generic tool to implement individual programs which are as diverse as the individual commands themselves. This tool is given to personnel who are not qualified to use it or even understand it. This seems to be an almost sure-fire formula for failure, and yet we are struggling valiantly to implement security programs within these constraints. Given these circumstances it seems important that we be able to filter the information presented in the generic tool in order to get a tool that is more applicable to the individual command. Some mechanism which does this seems to be an integral element to any true solution this problem.

The functionality and practicality of the system just proposed is largely dependent on developing technology. Current systems do not fully exploit the possibility of creating multiple views, "but this aspect will become more and more prominent as authors learn to create more dimensions for the user on the underlying information." [Bruza, and van der Weide, 1990] As the databases which underlie the hypertext

system grow larger, the need for successful information retrieval tools based on context will become more acute. Here successful is defined by the ability to achieve high recall and high precision. Although all of the described tools and characteristics discussed in this chapter might not be practical for the proposed system, they will surely be necessary for the very large hypertext databases of the future.

# VI. SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

## A. SUMMARY

Hypertext technology represents a potentially very powerful tool for certain applications within the Department of Defense. The goal of this thesis was to provide a good overview of hypertext technology to determine its feasibility for resolving some of the problems currently facing newly assigned and inexperienced ADP security officers throughout the Department of the Navy. This included what hypertext actually is, insight into the current capabilities, potential problems and limitations, and potential application areas for hypertext within the Department of Defense.

The thesis was organized in three basic parts. Chapter II identified some of the problems facing newly assigned ADP Security Officers. Specific issues and possible solutions were identified. Chapters III and IV gave an extensive overview of hypertext to include its capabilities, limitations, potential applications, and design issues. The last part of the thesis focused on the general application of hypertext within the Department of Defense and general design issues for a system which addressed the problem of ADP security. Chapter V used the background in Chapters III and IV to help identify broad areas of hypertext applications in the Department of Defense.

Chapter VI used the DON AIS Security Guidelines as an example of a potential application area to help frame design considerations for hypertext systems. It also evaluated the practicality of a hypertext solution to address some of the issues identified in Chapter II.

Although originally intended to simply address the problem of implementing an ADP security program, the thesis had more broad implications for other Department of Defense applications. This thesis provided the background information necessary to evaluate these potential applications in terms of practicality and usefulness.

## B. CONCLUSIONS

Our proclivity within DOD for using documents in virtually every facet of our work suggests that hypertext has an extremely bright future in the DOD. Chapter V identifies a number of general application areas which show great promise. The potential benefits from using hypertext systems are very real. They are comprised of benefits which both save money and help us perform our mission.

Although hypertext has enormous potential benefit, its application must be judicious. Great care must be taken to carefully analyze the problem domain and identify user needs to frame system design issues. Users in the Department of Defense are primarily task driven in their use of hypertext

systems. This has implications for a number of specific design considerations which were identified.

In most cases simply converting an existing document into a hypertext database has little value. Hypertext has a number of inherent problems which must be overcome or compensated for if the system is to gain user acceptance and have practical value. Usually something of value must be added to the system in terms of additional functionality or it must offer some other form of substantial improvement such as reduced storage costs or improved document updates.

The integration of hypertext and other technologies greatly extends the feasibility of hypertext as a solution to a number of problem domains within the DOD. The incorporation of artificial intelligence or expert systems with hypertext is extremely synergistic.

If hypertext is to be a practical tool for implementing an ADP security program in the Navy, it seems important that the information presented in the DON AIS Security Guidelines (Or whatever else the database is comprised of) is filtered to derive a tool more adaptable to an individual's command circumstance. Some mechanism which does this seems to be an integral element to any true solution to this problem.

Many of the applications within the Department of Defense will have huge hypertext databases. In order for hypertext to be practical for these applications, improvements in information retrieval techniques, navigation tools, and view

construction must be realized. Given the enormous potential of hypertext and the great deal of research interest in this area, these improvements are sure to come.

## C. RECOMMENDATIONS FOR FUTURE RESEARCH

There are a number of areas where research attention could be focused. The comparison of hypertext to other computer systems and to printed material offers numerous opportunities. Studies which evaluate hypertext systems and expert systems for the same application area would be beneficial to determine the best strategy for integrating these technologies. Any study which measures hypertext useability would be valuable. Currently there seems to be a prevailing attitude that hypertext of course offers improved system performance. This is not the case for all applications.

Research areas within the Department of Defense include the investigation of design issues for very large hypertext databases, the technological feasibility for application in the general areas suggested in Chapter V, prototyping of these systems to identify design and implementation problems, and cost benefit analysis to determine potential cost savings for specific applications.

## LIST OF REFERENCES


Akscyn, R.M., McCracken, D.L., and Yoder, E.A., "KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations," *Communications of the ACM*, v.31 no.7, pp. 820-835, July 1988.

Aischuler, Liora, "Hand-Crafted Hypertext-Lessons from the ACM Experiment," in Barrett, E. (ed): *The Society of Text*, MIT Press, 1989, pp. 343-361.

Begeman, Michael, L., and Conklin, Jeff, *Byte*, pp. 255-266, October 1988.

Bevan, S.J., and Wetherall, P.R., "The Conversion of an Expert System to an Intelligent CAI System," in Seideland, R.J. and Weddle, P.D.,(ed): *Computer Based Instruction in Military Environments*, Plenum Press, 1987, pp. 21-28.

Brockmann, J.R., Horton, W., and Brock, K., "From Database to Hypertext via Electronic Publishing: An Information Odyssey," in Barrett, E. (ed): *The Society of Text*, MIT Press, 1989, pp. 162-205.

Bruza, P.D., and van der Weide, T.P., "Assessing the Quality of Hypertext Views," *Sigir Forum*, v.24, no.3, ACM Press, pp. 6-25 September 10 1990.

Bush, Vannevar, "As We May Think," Atlantic Monthly, v.176, pp. 226-236, July 1945.

Campbell, Brad and, Goodman, Joseph, M., "HAM: A General Purpose Hypertext Abstract Machine," *Communications of the ACM*, v.31, no.7, pp. 856-861, July 1988.

Carlson, Patricia, A., "Hypertext and Intelligent Interfaces for Text Retrieval," in Barrett, E. (ed): *The Society of Text*, MIT Press, 1989, pp. 59-74.

Conklin, Jeff, "Hypertext: An Introduction and Survey," *IEEE Computer*, pp. 17-41, September 1987.

Coombs, James, H., "Hypertext, Full Text and Automatic Linking," in Vidick, J.L.(ed): *ACM SIGIR'90, Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, (5-7 September 1990), ACM Press, pp. 83-98.

Croft, Bruce, W., and Turtle, Howard, "The Use of Cluster Hierarchies in Hypertext Information Retrieval," *Proc. ACM Hypertext'89 Conference* (Pittsburgh, PA, 5-8 November), pp. 225-237.

Department of the Air Force, "Computer Based Instruction for Computer Systems Security Officers - An Example by the Air Force Cryptologic Support Center Kelly AFB, San Antonio, Texas," in Burrows, J.H. and Gallagher, P.R. Jr., (ed.): *12th National Computer Security Conference*(Baltimore, MD, 10-13 October 1989), pp. 505-514.

Department of the Navy, *Department of the Navy Automated Information Systems Security Program*, SECNAVINST 5239.2, 15 November 1989.

Department of the Navy, *Department of the Navy Automated Information Systems Security Guidelines*.

Department of the Navy, *CINCLANTFLT Shipboard ADP Security Program, an Approach*. CINCLANTFLT, code N742.

DeRose, Steven, J., "Expanding the Notion of Links," *Proc. ACM Hypertext'89 Conference* (Pittsburgh, PA, 5-8 November), pp. 249-257.

Evans, Gary J., *Identifying Security Problems and Devising Control Solutions in a Local Area Network:A Case Study Approach*, M.S. Thesis, Naval Postgraduate School, Monterey, California, September, 1990.

Fersko-Weiss, Henry, "3-D Reading with the Hypertext Edge," *PC Magazine*, v.10, no.10, pp. 241-267, May 28, 1991.

Fiderio, Janet, "A Grand Vision," *Byte*, pp. 237-244, October 1988.

Frisse, Mark, "From Text to Hypertext," *Byte*, pp. 247-253, October 1988.

Grice, Roger, A., "Online Information: What Do People Want? What Do People Need?," in Barrett, E. (ed): *The Society of Text*, MIT Press, 1989, pp. 22-44.

Halasz, Frank, G., "Reflections on Notecards: Seven Issues for the Next Generation of Hypermedia Systems," *Communications of the ACM*, v.31, no.7, pp. 836-852, July 1988.

Hayes, Phil, and Pepper, Jeff, "Towards an Integrated Maintenance Advisor," *Proc. ACM Hypertext'89 Conference* (Pittsburgh, PA, 5-8 November), pp. 119-127.

Herrstrom, David, S., and Massey, David, G., "Hypertext in Context," in Barrett, E. (ed): *The Society of Text,* MIT Press, 1989, pp. 45-58.

Jaynes, Joseph, T., "Limited Freedom: Linear Reflections on Nonlinear Texts," in Barrett, E. (ed): *The Society of Text,* MIT Press, 1989, pp. 148-161.

Jones, Tricia, "Towards a Typology of Educational Uses of Hypermedia," in Goos, G., and Hartmanis, J., (ed): *Lecture Notes in Computer Science, Computer Assisted Learning 3rd International Conference, ICCAL'90,* (Hagen, FRG, June 1990), pp. 265-276.

Jordan, Daniel, S., and others, "Facilitating the Development of Representations in Hypertext with IDE," *Proc. ACM Hypertext'89 Conference* (Pittsburgh, PA, 5-8 November), pp. 93-103.

Keen, Peter, G., "Decision Support Systems: The Next Decade," *Decision Support Systems,* v.3, pp. 253-265, 1987.

Kellet, Daniel, A., "*Hypertext: Another Step Toward the Paperless Ship,*" M.S. Thesis, Naval Postgraduate School, Monterey, California, June, 1989.

Kitfield, James, "Is Software DOD's Achilles' Heel?," *Military Forum,* pp. 28-35, July, 1989.

Lafferty, Larry, and others, "Techniques for capturing expert knowledge: an expert systems/hypertext approach," *Applications of Artificial Intelligence VIII,* SPIE Vol 1293, 1990, pp. 181-191.

MacDonald, Scot, "Paper--You Can't Shoot it at the Enemy," Surface Warfare, pp. 14-16, September/October 1987.

Marshall, Catherine, C., and Irish, Peggy, M., "Guided Tours and On-Line Presentations: How Authors Make Existing Hypertext Intelligible for Readers," *Proc. ACM Hypertext'89 Conference* (Pittsburgh, PA, 5-8 November), pp. 15-26.

Meyrowitz, Norman, "The Missing Link: Why We're All Doing Hypertext Wrong," in Barrett, E. (ed): *The Society of Text,* MIT Press, 1989, pp. 107-114.

Nielsen, Jakob, "The Matters That Really Matter for Hypertext Usability," *Proc. ACM Hypertext'89 Conference* (Pittsburgh, PA, 5-8 November), pp. 239-248.

Nielsen, Jakob, *Hypertext and Hypermedia*, Academic Press Inc., 1990a.

Nielsen, Jakob, "The Art of Navigating through Hypertext" *Communications of the ACM,* v.33 no.3, pp. 296-310, March 1990b.

Perlman, Gary, "Asynchronous Design/Evaluation Methods for Hypertext Technology Development," *Proc. ACM Hypertext'89 Conference* (Pittsburgh, PA, 5-8 November), pp. 61-82.

Raymond, Darrell, R., and, Tompa, Frank, WM., "Hypertext and the Oxford English Dictionary," *Communications of the ACM,* v.31, no.7, pp. 871-878, July 1988.

Rubens, Philip, "Online Information, Hypermedia, and the Idea of Literacy," in Barrett, E. (ed): *The Society of Text,* MIT Press, 1989, pp. 3-21

Shneiderman, Ben, "Reflections on Authoring, Editing, and Managing Hypertext," in Barrett, E. (ed): *The Society of Text,* MIT Press, 1989, pp. 115-131.

Smith, John, B., and Weiss, Stephen, F., "Hypertext," *Communications of the ACM,* v.31, no.7, pp. 816-819, July 1988.

Turban, Efraim, *Decision Support and Expert Systems,* Macmillan Publishing Company, 1990.

Van Dyke Parunak, H., "Hypermedia Topologies and User Navigation," *Proc. ACM Hypertext'89 Conference* (Pittsburgh, PA, 5-8 November), pp. 43-50.

Zellweger, Polle, T., "Scripted Documents: A Hypermedia Path Mechanism," *Proc. ACM Hypertext'89 Conference* (Pittsburgh, PA, 5-8 November), pp. 1-14.

Zviran, M., Hodge, J.C., and, Micucci, V.A., "SPAN-A DSS for Security Plan Analysis," *Computers and Security,* v.9, no.2, pp. 153-160, 1990.

## INITIAL DISTRIBUTION LIST

| | | |
|---|---|---|
| 1. | Defense Technical Information Center<br>Cameron Station<br>Alexandria, Virginia 22304-6145 | 2 |
| 2. | Library, Code 52<br>Naval Postgraduate School<br>Monterey, California 93943-5002 | 2 |
| 3. | Professor Tung Bui Code AS/Bd<br>Department of Administrative Sciences<br>Naval Postgraduate School<br>Monterey, California 93943-5000 | 1 |
| 4. | Professor Moshe Zviran Code AS/ZV<br>Department of Administrative Sciences<br>Naval Postgraduate School<br>Monterey, California 93943-5000 | 1 |